



Universidad
Carlos III de Madrid

Herramienta de ETL Powercenter Workflow Manager y Monitor

Autor: Mohamed Belhaj

Tutor: Miguel Ángel Patricio Guisado

1.	Introducción.....	4
1.1.	Tema del proyecto.....	4
1.2.	Motivación.....	6
1.3.	Objetivos.....	7
2.	Estado del arte.....	8
2.1.	Herramientas ETL.....	8
2.2.	Base de datos.....	12
2.3.	Control y programación de flujo de ejecución.....	14
2.4.	Informatica PowerExchange.....	14
3.	Gestión del proyecto.....	16
3.1.	Definición del Workflow Manager.....	16
3.2.	Tipos de tareas.....	17
3.3.	Ventanas del Workflow Manager.....	17
3.4.	Configuración del Workflow Manager.....	18
3.4.1.	Opciones generales.....	19
3.4.2.	Opciones formato.....	21
3.4.3.	Opciones diversas.....	21
3.4.4.	Opciones avanzadas.....	22
3.5.	Toolbars.....	23
3.6.	Búsqueda de objetos.....	24
3.6.1.	Búsqueda simultánea.....	24
3.6.2.	Búsqueda individual.....	25
3.7.	Organizar los objetos en el Workspace.....	25
3.8.	Control de versiones de objetos.....	25
3.8.1.	Check in.....	26
3.8.2.	Check out.....	27
3.8.3.	Consultar el histórico.....	27
3.8.4.	Comparar objetos.....	27
3.8.5.	Búsqueda de versiones.....	30
3.8.6.	Copiado de objetos de repositorio.....	32
3.9.	Metadatos.....	33
3.10.	Editor de expresiones.....	35

3.11. Conexiones.....	36
3.12. Workflow y Worklet.....	38
3.12.1. Definición del workflow.....	38
3.12.2. Configuración del Workflow.....	42
3.12.3. Definición de la Sesión.....	48
3.12.4. Configuración de la Sesión.....	49
3.13. Ejecución de workflows y sesiones.....	66
3.14. Importación y Exportación de Objetos.....	68
3.15. Workflow Monitor.....	69
3.15.1. Estado de ejecución.....	69
3.15.2. Propiedades de la ejecución.....	70
3.15.3. Log de la sesión.....	70
4. Implementación de casos prácticos.....	73
4.1. Introducción.....	73
4.2. Requisitos.....	73
4.3. Soluciones propuestas.....	74
4.4. Implementación.....	75
4.4.1. Fase 1.....	75
4.4.2. Fase 2.....	82
4.4.3. Fase 3.....	90
5. Conclusiones y trabajos futuros.....	98
5.1. Conclusiones.....	98
5.2. Trabajos futuros.....	99
6. Anexos.....	100
6.1. Coste del proyecto.....	100
6.2. Planning.....	101
6.3. Scripts Oracle.....	102
6.4. Scripts Shell.....	104
7. Bibliografía.....	107

1. Introducción

1.1. Tema del proyecto

El entorno que trata este proyecto es la inteligencia de negocio conocida como *Business Intelligence* (BI). Cuando hablamos de BI nos referimos a un conjunto de metodologías y soluciones tecnológicas destinadas a la ayuda de toma de decisiones de una empresa, permitiendo a los altos directivos controlar los negocios. Dicho de otra manera, BI es una solución que ayuda a comprender los mercados, clientes o proveedores con mayor claridad, captando y analizando los datos de un sector de negocio con el fin de mejorar los procesos de negocio y competir con más efectividad.

La noción de BI apareció a finales de los años 70 con los primeros info-centres. Sistemas que enviaban consultas directamente sobre los servidores de producción, lo que suponía un cierto riesgo. En los años 80, la llegada de las bases de datos relacionales y el modo Cliente-Servidor, ha permitido de aislar la informática de producción de los dispositivos decisionales.

Actualmente las empresas almacenan una gran cantidad de datos de diversos orígenes con el fin de obtener el mayor conocimiento posible. Sin embargo, lo cierto es que cuanto más datos se dispone, menos conocimiento tendremos. El valor de estos datos reside en la información que proporcionan, ya que pueden contener información oculta y necesaria que a simple vista no se logra apreciar, motivo por el que es necesario centralizar, estructurar y contextualizar estos datos para extraer valor de ellos a base de análisis profundo.

Se puede separar las metodologías del BI en dos fases: Back-end y Front-end. En esta última entran las tecnologías de *Reporting* que se acercan más al uso del usuario final, ofreciendo una presentación gráfica y entendible ayudando de esta manera a tomar decisiones. En la fase de Back-end, se encuentran las tecnologías de ETL que vamos a abordar en este proyecto.

Los procesos ETL (Extract, Transform & Load) de extracción, transformación y carga de datos aseguran la centralización de datos provenientes de diferentes orígenes en una base de datos denominada *Datawarehouse*. La primera fase de extracción consiste en leer los datos de las fuentes de información sin apenas aplicar cambios sobre los mismos. La transformación consiste en limpiar información incoherente, adaptar los tipos de datos, los valores de los campos y eventualmente cálculos. Por último la carga consiste en guardar los datos en el formato adecuado y de manera homogénea en el *Datawarehouse*.

Las herramientas ETL suelen guardar la información transformada en tablas relacionales con esquemas especiales. Estos procesos son automáticos, programables, monitorizables y contienen un riguroso control de errores para asegurar la fiabilidad de los datos cargados.

1.2. Motivación

El motivo por el que he decidido hacer este proyecto enfocado sobre la fase de ETL y concretamente en el sector bancario, es el contacto que he tenido con la herramienta de ETL Powercenter en varios proyectos bancarios.

Otro motivo ha sido la inmensa inversión que se realiza en este ámbito para hacer frente a las problemáticas encontradas en este sector, y es que no solo se trata de obtener información significativa en una masa de datos brutos con el fin de mejorar los servicios etc, sino encontrar soluciones a las amenazas que se dan en este sector.

La comunidad bancaria se enfrenta a varias amenazas: fraude, piratería, ciberataques, etc. Los bancos pueden facilitarse la tarea transformando los datos recopilados diariamente relativos a transacciones en análisis complejo haciendo intervenir la gestión de clientes, riesgos, relación cliente, rentabilidad de clientes, grado de fidelización clientela, evaluación de campañas de marketing y la tendencia del mercado.

En todos los proyectos en los que he colaborado, la herramienta de ETL utilizada ha sido Powercenter. Es una herramienta bien posicionada en el mercado y considerada como segura, robusta y escalable, satisfaciendo todas las necesidades de integración de datos. Cada proyecto tiene sus características y engloba diversas variables como pueden ser: volúmenes de datos, requisitos de latencia, infraestructura IT y metodologías. Powercenter varias ventajas sobre otras herramientas ETL en este sentido y cubre las exigencias que varían de un proyecto a otro.

1.3. Objetivos

El objetivo principal de este proyecto es explicar el modo de funcionamiento de las herramientas Workflow Manager y Monitor de Powercenter, describiendo todos sus componentes de manera a familiarizarnos con la herramienta y entender los primeros pasos a seguir para empezar a desarrollar. También se realizarán algunos procesos de ETL destinados a cubrir unos casos de uso que veremos en detalle.

Como se ha mencionado antes, Powercenter permite hacer extracciones de diferentes fuentes como, bases de datos, ficheros planos, ficheros cobol, etc. Aquí vamos a ver algunos ejemplos de carga de datos en dos o tres fases.

En primer lugar veremos un ejemplo de la primera fase de *Staging* que consiste en extraer datos de una fuente ya sea una base de datos transaccional o un fichero plano y cargarlos en una base de datos que forma el almacén operacional de datos ODS (*Operational Data Store*) utilizado como área lógica provisional de almacenamiento. Los procesos de esta fase suelen ser simples ya que no se realizan transformaciones sobre los datos más allá de algunos controles y conversiones de tipos de datos con el objetivo de extraer la información de la manera más rápida posible y liberar la base de datos fuente lo antes posible.

La segunda fase consistirá en construir procesos para cargar datos del ODS en una base de datos DWH (*Datawarehouse*) y procesos que extraerán datos del DWH para guardarlos en ficheros planos.

Por último veremos mecanismos de ejecución programada de estos procesos, haciendo uso de scripts de Unix y una herramienta específica.

2. Estado del arte

En este apartado veremos las diferentes tecnologías actuales utilizadas en el sector haciendo hincapié sobre sus principales características y las diferencias, ventajas e inconvenientes entre ellas.

El objetivo de esta memoria se limita a la herramienta de ETL Powercenter que vamos a defender, sin embargo citaremos otras tecnologías rivales, así como algunas tecnologías necesarias (Oracle, Control-M, etc).

2.1. Herramientas ETL.

Para una empresa que busca tratar volúmenes de datos a gran escala con el fin de convertirlos en información significativa para utilizarla en operaciones cotidianas, exigencias operacionales, análisis de datos, grandes extracciones, transformaciones y carga de datos (ETL), es fundamental utilizar la herramienta que más se adapte a esas exigencias.

Además de Powercenter, las herramientas más conocidas en el mercado son IBM InfoSphere DataStage, Talend, Pentaho y Oracle ODI. En este apartado veremos una comparación de estas plataformas para identificar diferencias y similitudes.

- **Informatica Powercenter:**

Fundado en 1993, Powercenter es una excelente plataforma de integración de datos considerada líder en el mercado de herramientas ETL. Actualmente tiene más de 2600 clientes entre ellos 100 de las grandes compañías. Distribución de clientes: 18% empresas pequeñas, 23% empresas medianas y un 58% de clientela corresponde a grandes empresas.

Una de las características más relevantes que tiene es la tecnología "Pushdown Optimization" que consiste en encargar la propia base de datos de realizar algunas operaciones y extraer los datos necesarios de manera más rápida y menos costosa.

Powercenter es una herramienta muy segura, estable y con buen rendimiento, proporciona la información en tiempo y de manera correcta accediendo e integrando datos de todos los formatos. Ofrece soluciones a sus clientes cubriendo las exigencias de cada proyecto con diferentes ediciones: Standard, Advanced, Big data, Real time, Cloud Edition y virtualización de datos.

Sus principales características son:

- Aumenta la agilidad en el manejo de datos críticos y reporta al negocio.
- Aprovecha y explota la potencia del Big data.
- Se conecta y extrae datos de redes sociales.
- Simplifica el desarrollo de servicios web.
- Aumenta la productividad con la gestión de meta-datos.
- Realiza pruebas de cambios y los actualiza 10 veces más rápido e incrementa la cobertura de pruebas validando los datos con la opción Informatica Data Validation.
- Identifica proactivamente riesgos de integración de datos.
- Mejora el acceso a Mainframe y el rendimiento.
- Refuerza las operaciones en tiempo real.

Cabe recordar que exige un buen nivel de conocimiento de SQL para realizar desarrollos y perfiles experimentados para su correcta instalación. Un inconveniente a tener en cuenta es el precio elevado de sus licencias.

- **IBM InfoSphere DataStage:**

Utiliza frameworks paralelos, integra datos a través de múltiples sistemas y soporta la gestión de meta-datos. La escalable plataforma ofrece más flexibilidad en términos de integración de todo tipo de datos, incluyendo Big data en plataformas distribuidas y mainframe.

Sus principales características:

- Soporta la colección, integración y transformación de grandes volúmenes de datos, partiendo de simples estructuras de datos a más complejas.
- Proporciona integración en tiempo real incluyendo conexión entre orígenes de datos y las aplicaciones.
- Ayuda a optimizar la utilización del hardware y prioriza tareas críticas.
- Ayuda a mejorar la velocidad, flexibilidad y eficacia de desarrollar, desplegar, actualizar y gestionar la infraestructura de integración de datos.

- **Pentaho:**

Es una herramienta de integración de datos open-source comercial que dispone de un producto llamado Kettle especializado en integración de datos. Diseñado con una interface gráfica salida y fácil de usar. Lanzado en 2001, tiene una comunidad de 13.500 usuarios registrados.

Sus principales características son:

- Es un motor de Java autónomo que trata procesos y tareas para mover datos de entre varias bases de datos diferentes y ficheros.
- Permite programar la ejecución de tareas.
- Permite realizar opresiones sobre servidores “esclavos” en otras máquinas.
- Diseñado con las siguientes funcionalidades de calidad de datos: A partir de su propia interfaz gráfica, permite escribir consultas SQL personalizadas, Javascript y expresiones regulares.

- **Talend:**

Es una herramienta de integración de datos open-source, pero no una plataforma completa de BI. Creada en 2006, es menos popular que Pentaho pero tiene dos entidades financieras que la patrocinan.

Sus principales características son:

- Utiliza un generador de código y una interfaz gráfica, pero únicamente dentro de Eclipse RC.
- Genera código Java o Perl que puede ser ejecutado en el servidor.
- Permite planificar tareas.
- Dispone de las siguientes funcionalidades de calidad de datos: A partir de su propia interfaz gráfica, permite escribir consultas SQL personalizadas y Java.

- **Oracle ODI:**

Considerada la mejor plataforma del mercado especializada en transferencia en masa de datos rápida y el manejo de transformaciones complejas de datos. Completamente integrada en el paquete tecnológico de Oracle. ODI sigue invirtiendo en una plataforma estratégica de integración de datos y suele estar incluida en el paquete de productos de Oracle para posicionarse en el mercado.

Sus principales características son:

- Dispone de un software de componentes modulares que funciona sobre una gama de plataformas populares.
- Permite operar y desplegar componentes que no forman parte del entorno Oracle y sobre múltiples servidores de aplicaciones.
- Soporta IBM WebSphere Support y IBM WebSphere Application Server.
- Diseñado con XML.

2.2. Base de datos.

En este proyecto hemos elegido una base de datos Oracle 11g ya que es una plataforma completa para almacenes de datos y Business Intelligence que combina rendimiento y escalabilidad líder en el mercado, análisis con alta integrabilidad y calidad de datos. Esta base de datos proporciona funciones excelentes para almacenes de datos *DWH* y almacenes de datos especializados *Datamart* con escalabilidad probada y rendimiento record. También proporciona una plataforma exclusivamente integrada para el análisis, incorporando OLAP, Data Mining y funciones estadísticas directamente en la base de datos.

- **Rendimiento:**

Esta base de datos incluye optimizaciones de rendimiento para cada tipo de entorno de DWH, teniendo en cuenta las cada vez más complejas cargas de trabajo de los almacenes de datos ya que a menudo hay diferentes usuarios que ejecutan operaciones muy diferentes con diferentes requerimientos para el rendimiento de consultas.

Para ello en esta BDD se incluyen las vistas materializadas basadas en OLAP.

- **Análisis:**

Oracle ofrece una gran ventaja al integrar OLAP, Data Mining y funciones estadísticas dentro del motor de base de datos. De esta manera evita mover los datos de un DWH a otros motores analíticos para realizar un mayor análisis.

- **Escalabilidad:**

En el mundo del BI, los almacenes de datos crecen exponencialmente, ya que cada vez hay más procesos automatizados recopilando más datos con mayor nivel de detalle. Oracle 11g ofrece capacidades que permiten la escalabilidad: Particionamiento, compresión, agrupamiento en clusters y paralelismo.

- **Particionamiento:**

Para administrar grandes bases de datos es esencial poder particionar las tablas para poder manejarlas mejor a medida que vayan creciendo. Los usuarios no suelen tolerar la latencia de la aplicación aun sabiendo que el volumen de datos crece cada vez más. El particionamiento permite agrupar datos según el patrón elegido en particiones separadas que posteriormente se puede acceder a ellas de manera más rápida sin tener que recorrer toda la tabla. Esto permite ahorrar mucho tiempo en obtener respuestas de consultas.

- **Compresión:**

El aumento casi diario de volúmenes de datos almacenados causa un gran problema de capacidad de almacenaje que deriva a su vez a un costo alto. Desde la versión Oracle 9i permite comprimir los datos almacenados en tablas relacionales. En esta versión se incluye una mejora que da soporte a actualizaciones e inserciones frecuentes y pequeñas.

- **Real Application Clusters:**

Permite que una base de datos escale a través de múltiples servidores. Para un DWH, esta tecnología ofrece una solución para escalar con el fin de aumentar a cientos de CPU, agregando de esta manera más potencia informática a un sistema de almacén de datos.

- Paralelismo:

Consiste en la capacidad de aprovechar varios procesadores en la ejecución de una sola operación de base de datos. La arquitectura de ejecución paralela y escalable se ajusta dinámicamente para maximizar el rendimiento general del sistema.

2.3. Control y programación de flujo de ejecución.

Los procesos necesitan seguir un orden lógico de ejecución establecido y se deben pisar entre ellos, por lo que es necesario disponer de un mecanismo para controlarlos. En función de las necesidades y envergadura de cada proyecto se suele utilizar una tecnología u otra, de las cuales citaremos la herramienta más utilizada Control-M.

Control-M es una herramienta que permite gestionar la integración de los procesos sin tener en cuenta la plataforma tecnológica donde reside la aplicación. Permite automatizar, programar y controlar el flujo de ejecución de los procesos ofreciendo una interfaz gráfica que facilita el trabajo.

En este proyecto hemos optado por utilizar Scripts Shell y *Crontab* de Unix para controlar el flujo y el orden de ejecución de los procesos de Powercenter, ya que no supone un costo adicional que supondría utilizar otra herramienta como Control-M.

2.4. Informatica PowerExchange.

En este proyecto veremos un caso de extracción de datos de una fuente de Cobol. Powercenter necesita conectores para “traducir” algunos formatos origen en un lenguaje entendible y así poder realizar el acceso y la entrega de la información (Access and Deliver). Este acceso y entrega deberían ser ejecutados con diferentes latencias, para cubrir estas necesidades la plataforma Informatica ofrece PowerExchange.

Este producto ofrece varios conectores para las diferentes aplicaciones, bases de datos o repositorios de datos. La licencia de cada conector se debe comprar por separado, sin embargo una vez adquirido se puede utilizar sobre tantas fuentes y destinos como queramos y no hay limitación de volumetría de datos ni número de procesadores por ejemplo. Algunos fabricantes licencian los conectores por volumen de datos, número de orígenes o destinos, procesadores, etc.

Algunos de los conectores más utilizados que ofrece:

- Bases de datos y DWH:
Oracle, SQL Server, Sybase ASE, Sybase IQ, Teradata, etc.
- Mainframes:
Flat files binarios, IDMS, IMS, VSAM, DB2 Linux, etc.

3. Gestión del proyecto

3.1. Definición del Workflow Manager

El Workflow Manager es la herramienta donde se crean y se gestionan los Workflows construidos en base a los mappings desarrollados en el Designer. Un Workflow es un conjunto de instrucciones que determinan como ha de ejecutarse el proceso. Éste debe estar formado, como mínimo, por una sesión y puede contener varias sesiones. Cada sesión se corresponde con un solo Mapping y contiene los parámetros de ejecución.

Un Workflow puede estar formado, además de sesiones, de comandos y otros objetos para gestionar las dependencias entre Workflows, para enviar correos electrónicos, o para gestionar las condiciones de encadenamiento entre sesiones.

También es posible crear un Worklet en el Workflow Manager. Un Worklet es un objeto que agrupa un conjunto de tareas, similar a un Workflow pero sin información de programación y es posible incluir varios en un solo Workflow.

Para crear un Workflow es necesario crear las tareas que lo van a componer, estas tareas pueden ser sesiones construidas sobre mappings o de otro tipo como se verá más adelante. Para ello, citaremos los espacios de trabajo donde se llevan a cabo estas operaciones:

- **Task Developer:** Espacio donde se crean las tareas que se ejecutarán en el Workflow.
- **Workflow Designer:** Espacio donde se crean los Workflows que se componen de una o una combinación de tareas conectadas entre sí.
- **Worklet Designer:** Espacio donde se crean los worklets.



Figura 1: Espacios de trabajo del WM.

3.2. Tipos de tareas

Las tareas que se pueden crear en esta herramienta son las siguientes:

- **Assignment.** Asigna un valor a una variable de workflow.
- **Command.** Contiene línea de comando Shell para ser ejecutada durante la ejecución del workflow.
- **Control.** Parar o abortar el workflow.
- **Decision.** Contiene una condición de decisión.
- **Email.** Manda un correo electrónico durante la ejecución del workflow.
- **Event-Raise.** Notifies the Event-Wait task that an event has occurred.
- **Event-Wait.** Espera la ocurrencia de un evento para ejecutar la siguiente tarea.
- **Session.** Se construye sobre un mapping existente para ser ejecutado.
- **Timer.** Fecha y hora programada para la ejecución.

3.3. Ventanas del Workflow Manager

El Workflow Manager tiene las siguientes ventanas para crear y organizar workflows:

- **Navigator.** Navegador donde se puede conectar a múltiples repositorios y carpetas. En esta ventana, el Workflow Manager muestra un icono rojo sobre los objetos inválidos.
- **Workspace.** Espacio de trabajo donde se puede crear, modificar y visualizar tareas, workflows y worklets.
- **Output.** Contiene pestañas para mostrar diferentes tipos de mensajes de salida. La ventana de resultados contiene las siguientes pestañas:
 - **Save.** Muestra mensajes cuando se salvaguarda un workflow, worklet o tarea. Por ejemplo, el resumen de la validación cuando se guarda un workflow o un worklet.
 - **Fetch Log.** Muestra mensajes cuando el Workflow Manager recupera los objetos del repositorio.
 - **Validate.** Muestra mensajes al validar un workflow, worklet o tarea.

- **Copy.** Muestra mensajes cuando se copian objetos del repositorio.
- **Server.** Muestra los mensajes del Servicio de Integración.
- **Notifications.** Muestra los mensajes del servicio de repositorio.
- **Overview.** Ventana opcional que permite explorar grandes workflows en el espacio de trabajo.

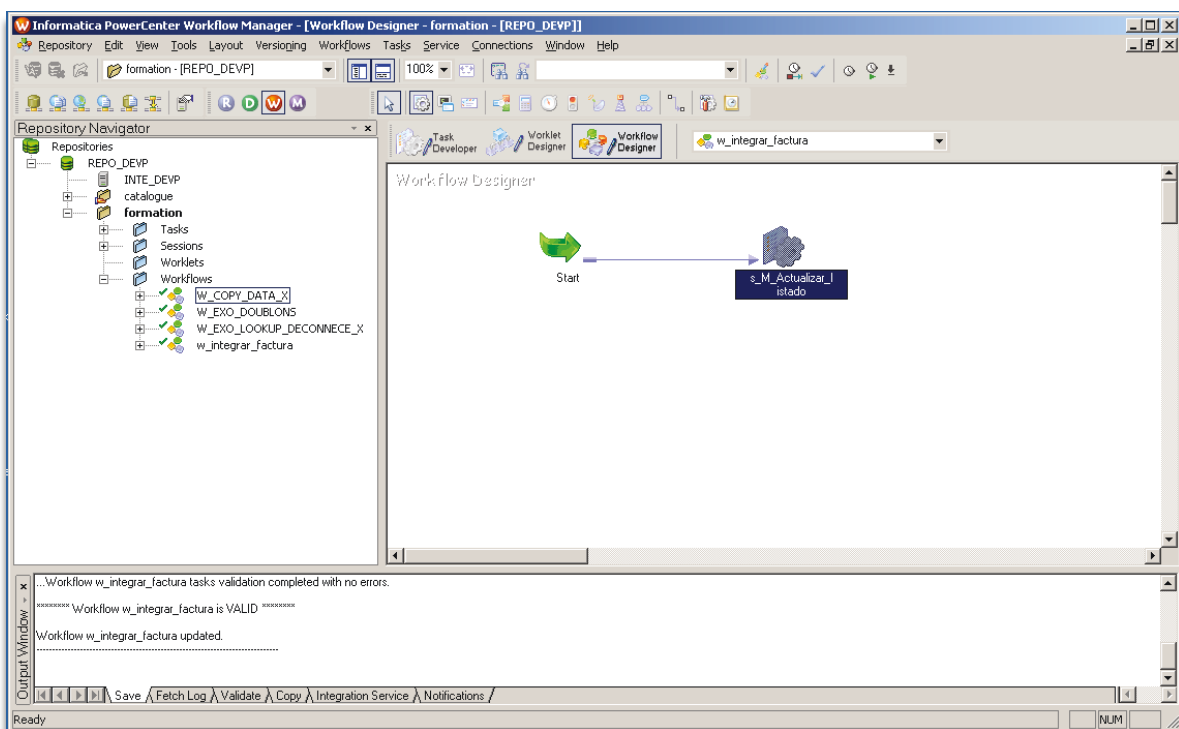


Figura 2: Ventanas del Workflow Manager

3.4. Configuración del Workflow Manager

Puede personalizar las opciones predeterminadas para controlar el comportamiento y la apariencia de las herramientas del Workflow Manager.

Para configurar el Workflow Manager, accede a: Tools > Options.

- **General.** Se puede configurar el espacio de trabajo, opciones de ventana y otras opciones generales.
- **Format.** Se puede configurar el fondo, color y otras opciones de formato.
- **Miscellaneous.** Se puede configurar las copias y el control de versiones.
- **Advanced.** Se puede configurar la seguridad de conexión de objetos.

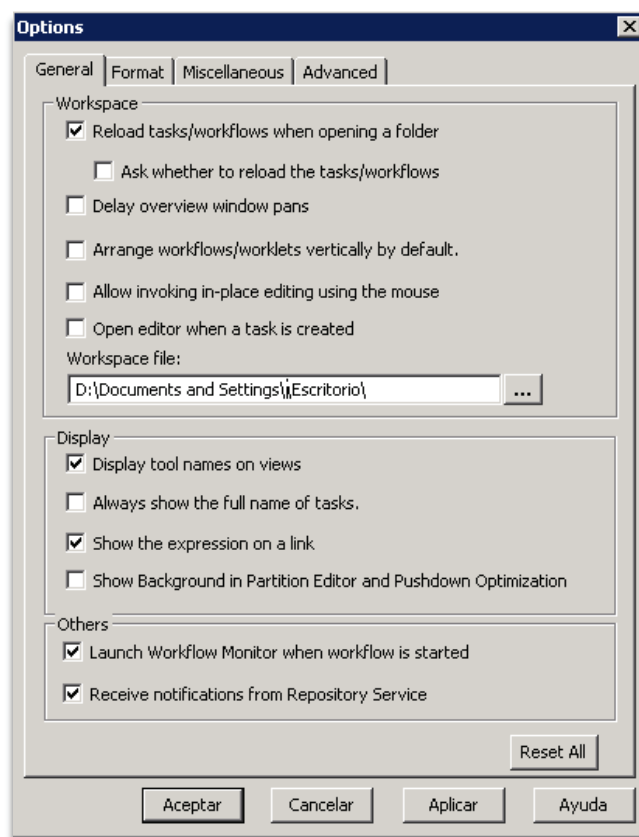


Figura 3: Opciones del Workflow Manager

3.4.1. Opciones generales

Son las opciones generales del comportamiento de la herramienta, como por ejemplo cuando ésta conserva su vista cuando se cierra, cómo se comporta la ventana general y de donde carga el Workflow Manager los archivos. A priori no parecen muy interesantes pero son bastante prácticas a la hora de desarrollar.

Opción	Descripción
Reload Tasks/Workflows When Opening a Folder	Vuelve a cargar la última vista de una herramienta al abrirlo. Por ejemplo, si un workflow está abierto cuando se desconecta de un repositorio, seleccione esta opción para que el mismo flujo de trabajo aparezca la próxima vez que abra la carpeta y Workflow Designer. Por defecto está activada.

Ask Whether to Reload the Tasks/Workflows	Aparece al seleccionar la opción Reload task/workflow al abrir una carpeta. Seleccionar esta opción si se desea que el Workflow Manager vuelva a cargar tareas, workflows y worklets cada vez que abra una carpeta. Por defecto está desactivada.
Delay Overview Window pans	Por defecto, al arrastrar el foco de la ventana Overview, éste se mueve al mismo tiempo. Al seleccionar esta opción, el foco del espacio de trabajo no cambia hasta que suelte el botón del ratón. Por defecto está desactivada.
Arrange Workflows/ Worklets Vertically By Default	Organizar, por defecto, las tareas en el workflow de manera vertical. Por defecto está desactivada.
Allow Invoking In-Place Editing Using the Mouse	De forma predeterminada, puede pulsar F2 para editar objetos directamente en el espacio de trabajo en lugar de abrir el cuadro de diálogo Editar tarea. Seleccione esta opción para poder hacer clic en el nombre del objeto en el espacio de trabajo para editar el objeto. Por defecto está desactivada.
Open Editor When a Task is created.	Abre el cuadro de diálogo Editar tarea al crear una tarea. De forma predeterminada, el Workflow Manager crea la tarea en el área de trabajo. Si no habilita esta opción, haga doble clic en la tarea para abrir el cuadro de diálogo Editar tarea. Por defecto está desactivada.
Workspace File Directory	Directorio de archivos del espacio de trabajo creados por el Workflow Manager. Los archivos de área de trabajo mantienen la última tarea o flujo de trabajo que se guardó. Este directorio debe ser local en el cliente de PowerCenter para evitar la corrupción de archivos o sobrescritos por varios usuarios. De forma predeterminada, el Administrador de flujo de trabajo crea archivos en el directorio de instalación del cliente PowerCenter.
Display Tool Names on Views	Muestra el nombre de la herramienta en la esquina superior izquierda del área de trabajo. Por defecto está activada.
Always Show the Full Name of Tasks	Muestra el nombre completo de la tarea cuando es seleccionada. Por defecto, el Workflow Manager muestra una abreviatura del nombre de una tarea en el espacio de trabajo. Esta opción está desactivada por defecto.
Show the Expression on a Link	Muestra la condición del enlace en el espacio de trabajo. Cuando esta desactivada, el Workflow Manager muestra una abreviatura de la condición del enlace. Por defecto esta activada.
Show Background in Partition Editor	Muestra el color de fondo de los iconos de objetos. Por defecto esta desactivada.

and Pushdown Optimization	
Launch Workflow Monitor when Workflow Is Started	Abre el Workflow Monitor al ejecutar una tarea o workflow. Por defecto esta activada.
Receive Notifications from Repository Service	Se pueden recibir mensajes de notificación en el Workflow Manager y visualizarlos en la ventana Output window. Las notificaciones incluyen información sobre los objetos que crean, modifican o eliminan otros usuarios. De la misma manera se reciben notificaciones sobre sesiones, workflows y worklets. El Servicio de repositorio notifica los cambios para indicar que los objetos pueden estar no actualizados. Para recibir la notificación, la carpeta que contiene el objeto debe estar abierta en el navegador, y el objeto debe estar abierto en el espacio de trabajo. También se reciben notificaciones de creación de usuarios por el gestor de SR. Esta opción esta activada por defecto.
Reset All	Resetear todas las configuraciones de formato.

3.4.2. Opciones formato

Las opciones de formato controlan los colores del espacio de trabajo y fuentes. Se pueden configurar para cada herramienta del Workflow Manager.

No vamos a detallar esta parte ya que no es importante.

3.4.3. Opciones diversas

Se trata de ajustes de copia y control de versiones de objetos así como modos de carga de datos.

Opción	Descripción
Validate Copied Objects	Valida los objetos copiados. Activada por defecto.
Generate Unique Name When Resolved to "Rename"	Genera nombres únicos para los objetos copiados si se selecciona la opción Rename. Por ejemplo, si el workflow wf_compras tiene el mismo nombre que otro workflow en el fichero destino, la opción Rename genera un único nombre wf_compras1. Por defecto esta activada.

Get Default Object When Resolved to "Choose"	Usa el objeto con el mismo nombre en el fichero destino si seleccionamos la opción Choose. Por defecto esta desactivada.
Show Check Out Image in Navigator	Muestra el icono de Check Out el objeto ha sido almacenado/liberado. Esta activada por defecto.
Allow Delete Without Checkout	Se puede eliminar versiones de objetos de repositorio sin hacer un check out previo. Sin embargo, no puede eliminar otros objetos que otros usuarios han hecho check out y seguramente estén modificando. Cuando se selecciona esta opción, el Servicio de Repositorio realiza un check out de nuestro objeto cuando lo eliminamos.
Check In Deleted Objects Automatically After They Are Saved	Realiza un Check In de los objetos eliminados después de una salvaguarda de los cambios en el repositorio. Cuando se quita esta opción, el objeto eliminado permanece fuera (sin Check out) y habría que hacer un Check In desde la vista de resultados. Esta opción esta desactivada por defecto.
Target Load Type	Restablece el modo de carga predefinido de las sesiones. Se puede elegir entre el modo Normal y Bulk. Cualquier cambio que se realiza hace efecto después del reinicio del Workflow Manager. Se puede sobrescribir estos ajustes en las propiedades de la sesión. Esta opción esta desactivada por defecto.
Reset All	Resetea todos los ajustes en valores predefinidos.

3.4.4. Opciones avanzadas

El Workflow Manager (WM) tiene una opción de seguridad avanzada para especificar los derechos por defecto que deberían tener las conexiones de objetos. Cuando está seleccionada el WM asigna los derechos predefinidos en conexiones de objetos, para usuarios, grupos y otros.

Si desactivamos esta opción, el WM asigna permisos de lectura, escritura y ejecución a todos los usuarios que deberían tener estos permisos en el grupo por defecto.

Para activar esta opción:

1. Click en Tools > Options.
2. Click en Advanced Tab.

3. Seleccionar Enable Enhanced Security.
4. Click OK.

3.5. Toolbars

En el Toolbars del Workflow Manager se pueden encontrar las siguientes funcionalidades:

- **Standard:** Contiene los botones para conectarse o desconectarse a los archivos del repositorio.
- **Connections:** Contiene botones para crear, editar las conexiones y asignar SI.
- **Repository:** Contiene botones para conectarse a las carpetas de los repositorios, importar y exportar objetos y gestionar los cambios.
- **View:** Personalizar la barra de herramientas y visualizar las propiedades de los objetos.
- **Layout:** Organizar, restaurar, buscar y hacer zoom sobre los objetos del espacio de trabajo.
- **Tasks:** Crear tareas.
- **Workflow:** Editar las propiedades de un workflow.
- **Run:** Ejecutar una sesión o workflow, o programar la ejecución del mismo.
- **Versioning:** Realizar check de objetos, deshacer un checkout, comparar versiones, listar los objetos de un checkout y las consultar de un repositorio.
- **Tools:** Permite conectarse a otras aplicaciones Cliente de Powercenter. Cuando se conecta a otra aplicación Cliente desde aquí, Powercenter utiliza la misma conexión para conectarse al repositorio y abre las mismas carpetas.

3.6. Búsqueda de objetos

El Workflow Manager incluye funciones de búsqueda para encontrar tareas, enlaces, variables, eventos en el área de trabajo, y el texto en la ventana de resultados. Puede buscar elementos en cualquier herramienta de Workflow Manager o en la ventana de salida.

Hay dos formas de buscar elementos en el espacio de trabajo:

- **Find in Workspace.**
- **Find Next.**

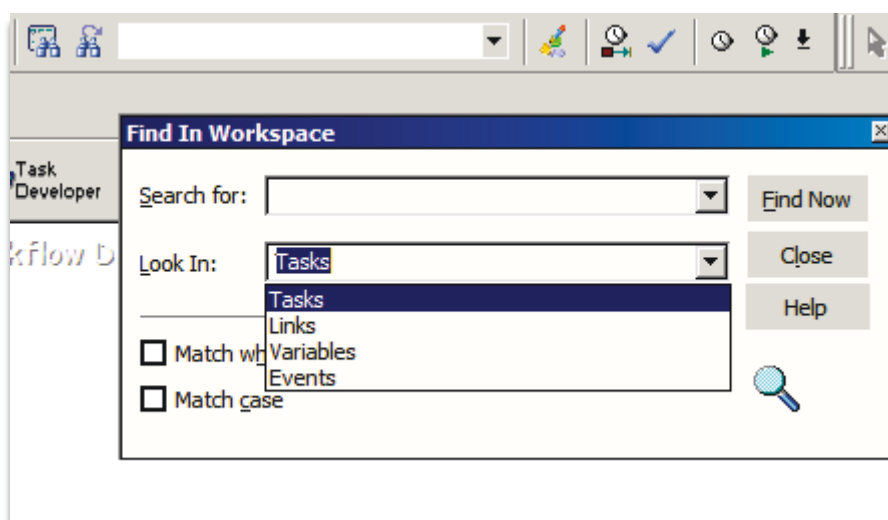


Figura 4: Búsqueda de objetos en el espacio de trabajo

3.6.1. Búsqueda simultánea

Se puede buscar varios elementos a la vez y devolver una lista de todos los nombres de las tareas, las condiciones de enlaces, nombres de eventos, o nombres de variables que contienen la cadena de búsqueda.

1. En cualquier herramienta del Workflow Manager, hacer clic en el botón *Find in Workspace* del Toolbar o haga clic en *Edit > Find in Workspace*.
2. Seleccionar la búsqueda de tareas, enlaces, variables o eventos.

3. Introducir una cadena de búsqueda o seleccione una cadena de la lista.

El Workflow Manager guarda los últimos 10 cadenas de búsqueda en la lista.

5. Clic en Buscar ahora.

3.6.2. Búsqueda individual

Al realizar una búsqueda de elementos de uno en uno, el Workflow Manager destaca el primer enlace, evento, variable o cadena de texto que contiene la cadena de búsqueda. Si repite la búsqueda destaca el siguiente elemento hasta encontrar todas las correspondencias. No se distingue entre mayúsculas y minúsculas.

3.7. Organizar los objetos en el Workspace

Cuando se trabaja sobre un workflow de gran tamaño, se necesita organizar los objetos para visualizar mejor los flujos. El Workflow Manager puede organizar los objetos en el espacio de trabajo horizontal o verticalmente.

En el Administrador de tareas, también puede organizar tareas uniformemente en el espacio de trabajo por la elección de la opción *Tile*. Para organizar los objetos en el espacio de trabajo, hacer clic en Layout> Arrange y seleccione Horizontal, Vertical, o Mosaico. Para visualizar los enlaces como líneas horizontales y verticales, hacer clic en Layout > Orthogonal Links.

3.8. Control de versiones de objetos.

Cuando se trabaja con objetos versionados, se debe desproteger el objeto haciendo un check out del mismo para poder modificarlo y guardarlo cuando desea confirmar los cambios en el repositorio. Es imprescindible volver a realizar el Check in para permitir que otros usuarios puedan recuperar el objeto y poder realizar nuevos cambios. Cada vez que se realiza un check in se añade una nueva versión del objeto en el repositorio.

3.8.1. Check in.

Cuando se realizan cambios, creación o modificación de objetos, éstos toman efecto cuando se realiza el Check in de los mismos en el repositorio, este último crea una nueva versión del mismo.

Nótese que incluso la supresión de un objeto no toma efecto hasta que no se haya hecho el Check in. De lo contrario podría causar conflictos entre usuarios que pueden visualizar un objeto ya suprimido e inutilizable por otro usuario.

Esto se aplica a todo tipo de objetos: Workflow, worklet, sesión o tarea.

Se puede hacer el check in de cada objeto por separado, seleccionando el objeto y haciendo click botón derecho sobre el objeto y marcando *Versioning > Check in*, o a través del Toolbar: *Versioning > Check in*. La opción *Find Checkouts* es muy útil para encontrar todos los objetos sobre los que un usuario está realizando cambios y hacer un Check in de todos a la vez.

Para deshacer cambios realizados sobre un objeto y mantener la versión anterior existente en el repositorio, se realiza un *Undo Checkout*.

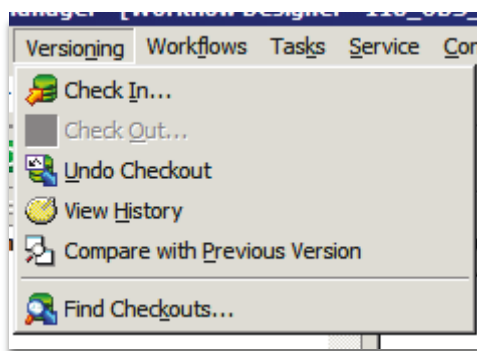


Figura 5: Versioning de objetos

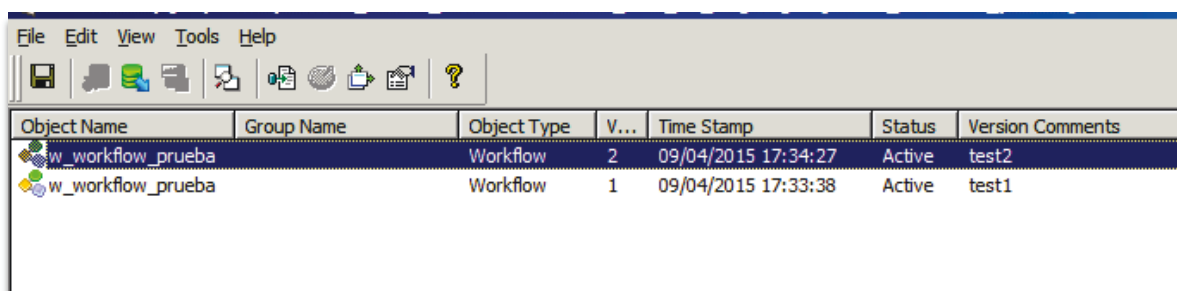
3.8.2. Check out.

Un objeto guardado en el repositorio (en check in) sólo lo puede recuperar un único usuario al mismo tiempo para poder realizar cambios sobre el mismo. Se puede realizar el Check out haciendo clic derecho sobre el objeto o en el menú como se muestra en la captura anterior.

3.8.3. Consultar el histórico

El workflow Manager ofrece la posibilidad de consultar el histórico de cambios realizados sobre los diferentes objetos del repositorio, proporcionando información sobre los cambios como: fecha, versión, usuario, etc.

Opción: Versioning > View History.



Object Name	Group Name	Object Type	V...	Time Stamp	Status	Version Comments
w_workflow_prueba		Workflow	2	09/04/2015 17:34:27	Active	test2
w_workflow_prueba		Workflow	1	09/04/2015 17:33:38	Active	test1

Figura 6: Consulta del histórico de cambios

3.8.4. Comparar objetos

Comparar dos objetos del repositorio del mismo tipo para identificar las diferencias entre los objetos. Por ejemplo, si tiene dos tareas Email similares en una carpeta, puede compararlos para ver cuál contiene los atributos que necesita. Cuando se comparan dos objetos, el Workflow Manager muestra sus atributos en detalle.

Puede comparar los objetos a través de las carpetas y los repositorios. Se deben abrir ambas carpetas para comparar los objetos. Puede comparar un objeto reutilizable con un objeto no reutilizable. También puede comparar dos versiones de un mismo objeto.

Se puede comparar los siguientes tipos de objetos:

- **Tasks**
- **Sessions**
- **Worklets**
- **Workflows**

Es de mucha utilidad la posibilidad de comparar las instancias del mismo tipo de objeto. Por ejemplo, si los workflows a comparar contienen instancias de un worklet con el mismo nombre, se pueden detectar diferencias entre las dos instancias. Nótese los siguientes casos:

- Instancias de sesiones y tareas en una comparación de workflow o worklet.
- Instancias de mapeos y transformaciones en una comparación sesión.
- Atributos de las instancias del mismo tipo dentro de una comparación de mapeo. Por ejemplo, cuando se comparan las fuentes de archivos planos, puede comparar los atributos, como el tipo de archivo (delimitado o fijo), delimitadores, caracteres de escape, y cita opcionales.

Para comparar versiones del mismo objeto:

Opción: Versioning > Compare with Previous Version.

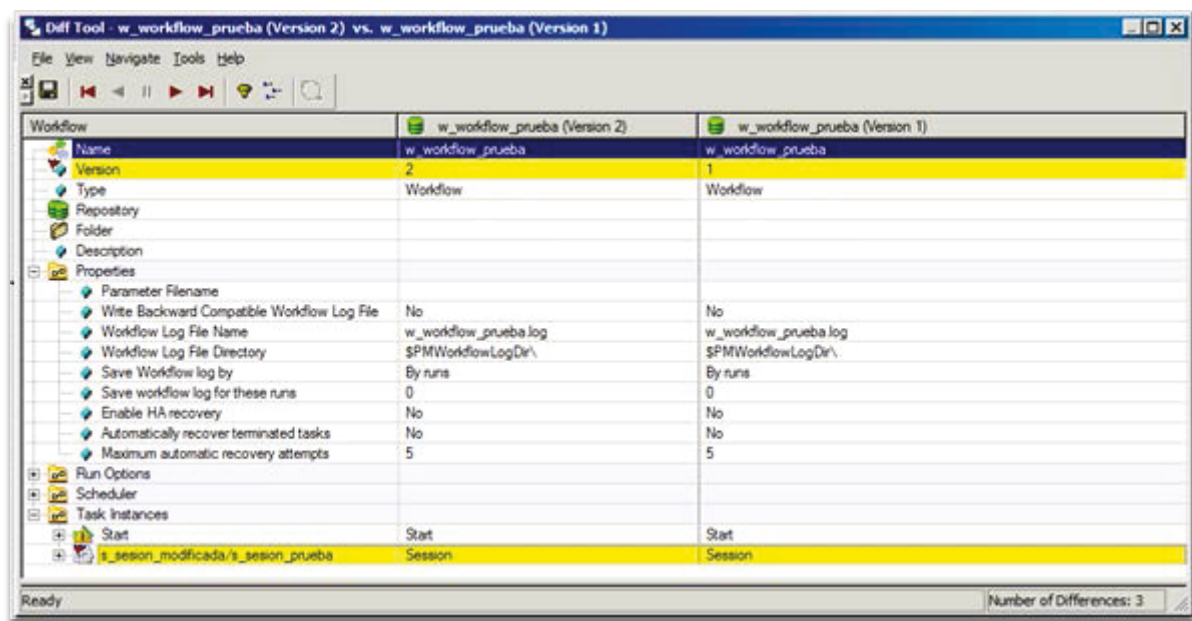


Figura 7: Comparar versiones de un objeto

Para comparar dos objetos:

El siguiente procedimiento compara objetos del mismo tipo, potencialmente, para detectar diferencias en la configuración.

- Abrir la carpeta que contiene el objeto a comparar.
- Abrir la ventana apropiada para el tipo de objeto seleccionado.
- Clic Task > Compare o Worklet > Compare o Workflow > Compare
- Seleccionar el objeto a comparar.
 - Se puede realizar la misma operación tanto en el navegador o el espacio de trabajo, seleccionando el objeto con el botón derecho y clic sobre Compare.
- El resultado de la comparación se puede guardar en fichero de texto o fichero HTML.

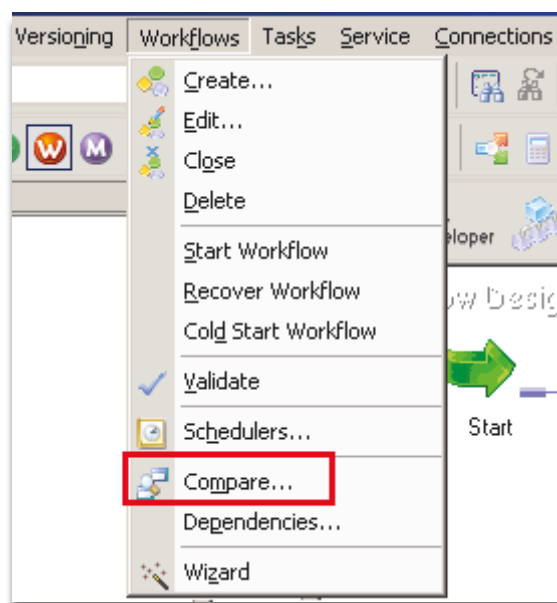


Figura 8: Comparar dos workflows

3.8.5. Búsqueda de versiones

Para buscar versiones de un objeto se utiliza una consulta sobre objetos en el repositorio que cumplen con las condiciones especificadas. Cuando se ejecuta una consulta, el repositorio devuelve únicamente los resultados que cumplen las condiciones descartando otras versiones del mismo objeto buscado. Estas consultas se pueden realizar para las siguientes tareas:

- Seguir objetos del repositorio durante el desarrollo. Usted puede agregar la etiqueta, Usuario, Última guardado o Comentarios parámetros a consultas para seguir objetos durante el desarrollo.
- Asociar una consulta con un grupo de despliegue. Cuando se crea un grupo de distribución dinámico, se puede asociar una consulta a este último.

Para crear una consulta de objeto, clic en *Tools > Queries* para abrir el *Query Browser*.

Desde el Query Browser, puede crear, editar y eliminar consultas. También puede configurar los permisos para cada consulta desde el Query Browser. Puede ejecutar cualquier consulta siempre y cuando se disponga de los permisos necesarios para el usuario.

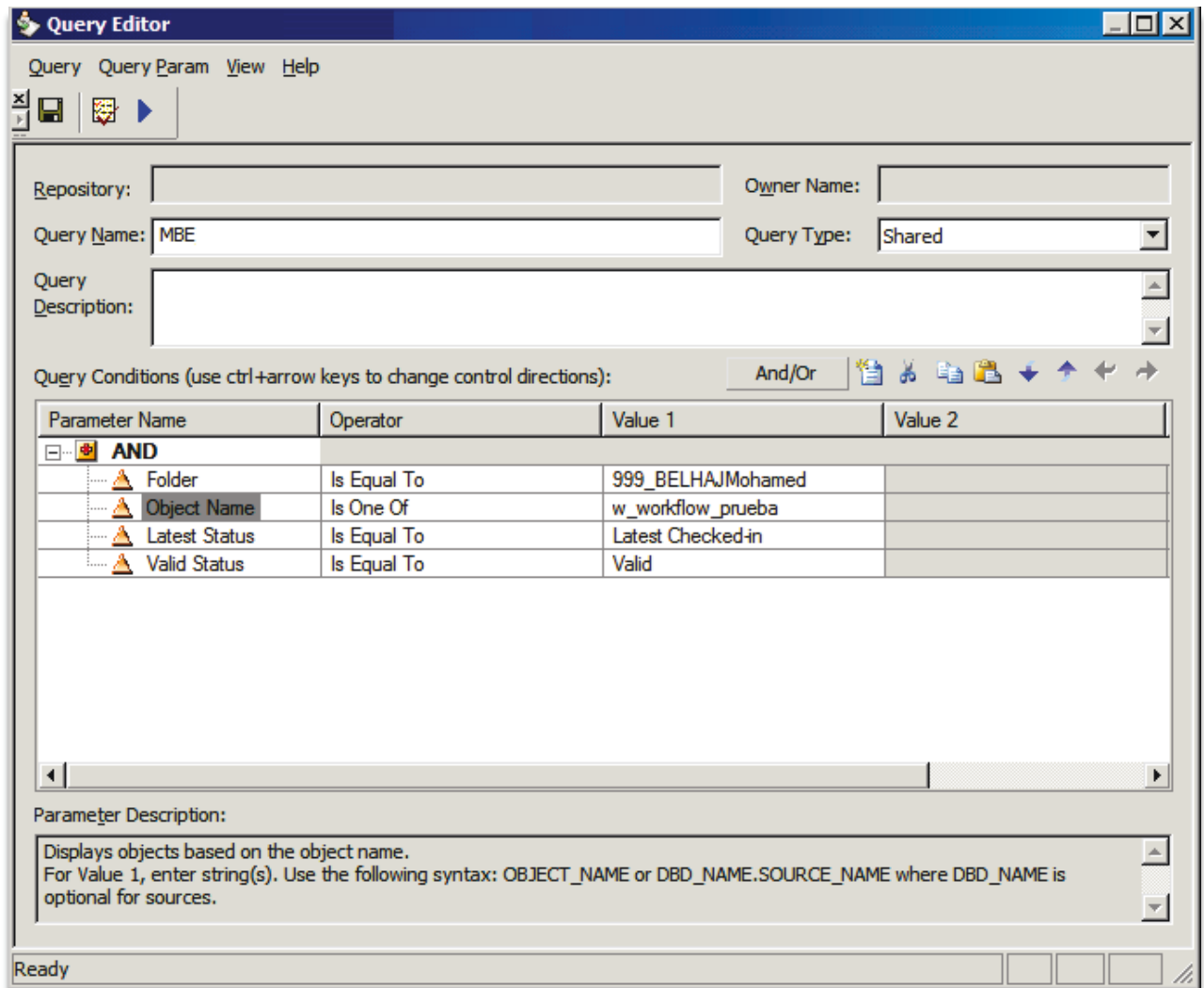


Figura 9: Búsqueda de versiones

3.8.6. Copiado de objetos de repositorio

Se puede copiar objetos del repositorio, como workflows, worklets o tareas dentro de la misma carpeta, a una carpeta diferente o en un repositorio diferente. Si queremos copiar el objeto a otra carpeta, se debe abrir la carpeta de destino antes de copiar el objeto en la carpeta (Conectarse a una carpeta no es lo mismo que desplegar para explorar).

Cuando se copia un workflow o un worklet, el asistente copia todos los worklets, sesiones y tareas que contiene. A veces se producen conflictos cuando el asistente encuentra un workflow o worklet con el mismo nombre en la carpeta de destino o cuando la conexión del objeto no existe en el repositorio de destino. Si la conexión no existe, puede omitir el conflicto y reconfigurarla después de copiar workflow ya que no es posible copiar las conexiones. Estos conflictos también pueden ocurrir al copiar una sesión.

Copiar una sesión:

Cuando se copia una sesión, el asistente busca la conexión de la BDD asociada a la carpeta destino. Si el mapping o la conexión no existen, es imprescindible elegir de nuevo el mapping o la conexión. En caso de que la carpeta destino con contenga ningún mapping, primero habrá que copiar el mapping en el Designer antes de copiar la sesión.

Copiar segmentos de un workflow:

Se puede copiar segmentos de workflows y worklets cuando se quiera reutilizar algunos componentes de los mismos. Un segmento consiste en una o varias tareas, los enlaces entre tareas y las condiciones que contienen. Se puede copiar objetos reutilizable y no reutilizable entre workflows de la misma carpeta o diferentes carpetas.

3.9. Metadatos

Se puede ampliar los metadatos almacenados en el repositorio asociando individualmente información con objetos del repositorio. Por ejemplo, es posible que sea necesario guardar el nombre del usuario con los worklets que ha creado, o añadir más información del mismo en una sesión.

En la siguiente tabla se describen las opciones de configuración de metadatos:

Opción	Descripción
Extension Name	Nombre del metadato a crear. Deben ser únicos para cada tipo de objeto en el mismo dominio, y no pueden contener caracteres especiales excepto "_" Y no pueden empezar por números.
Datatype	Tipo de datos: numérico (integer), string, boolean, o XML.
Value	Numérico: Integer. Boolean: Verdadero o falso. Para string o XML, clic sobre el botón de Edit para introducir el valor. El Workflow Manager no valida la sintaxis de XML.
Precision	Longitud máxima del tipo string o XML.
Reusable	Hacer que el metadato sea reusable o no-reusable para todos los objetos del mismo tipo seleccionado esta opción. Nota: Una vez creado como reusable, ya no se puede hacerlo no-reusable.
UnOverride	Esta columna aparece si uno de los metadatos ha cambiado. Para restaurar el valor por defecto, clic sobre Revert.
Description	Descripción del metadato creado.

Creación de metadatos.

A través del Workflow Manager se puede crear un metadato reusable, no-reusable o de un usuario definido para un objeto del repositorio. Editar el objeto para el cual se quiere crear el metadato añadiendo la información como se explica a continuación:

- Abrir el objeto en la ventana correspondiente del Workflow Manager.
- Clic en la pestaña Metadata Extensions.
 - Aquí se listean los metadatos existentes
- Clic en el botón de añadir.
- Configurar el metadato a añadir y clic OK.

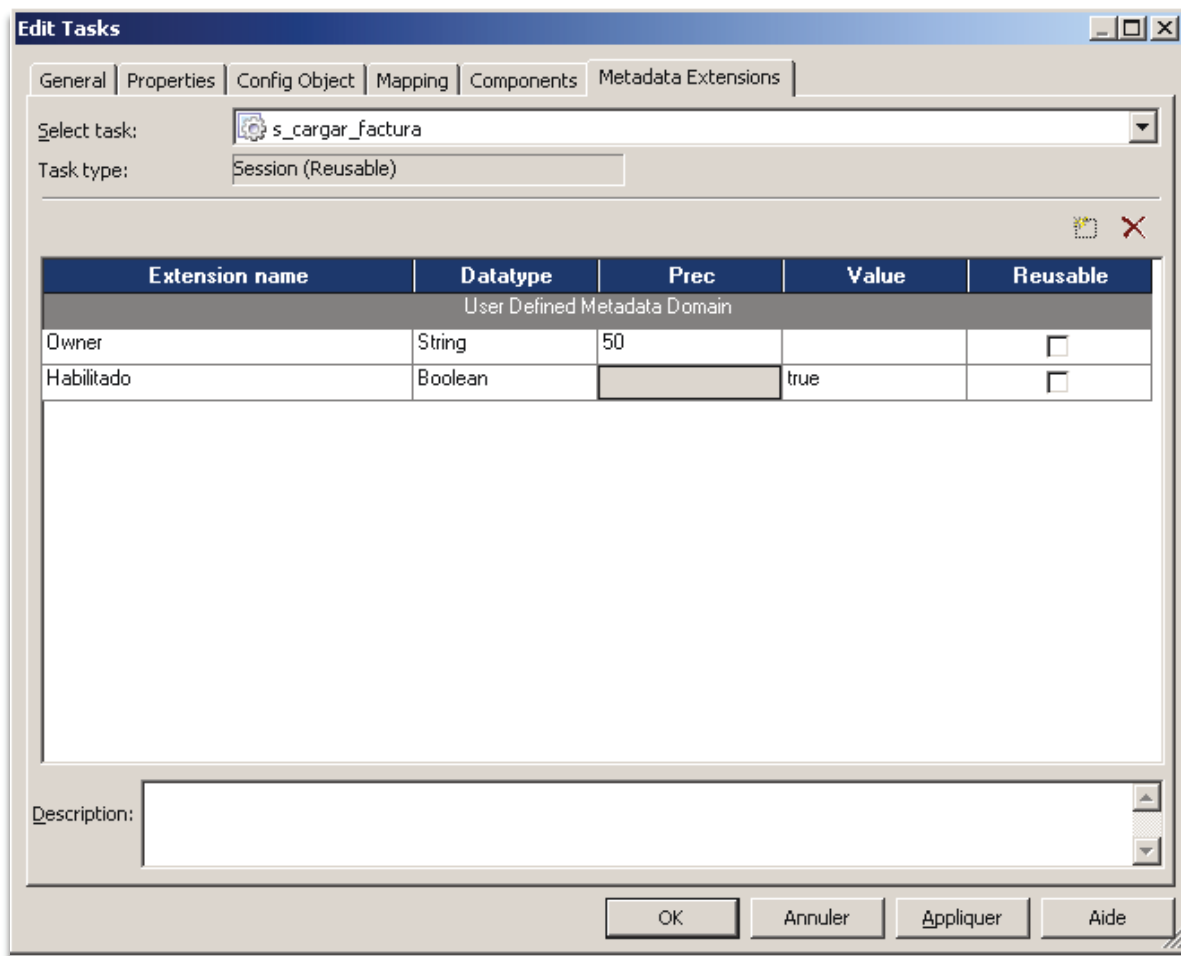


Figura 10: Creación de metadatos

3.10. Editor de expresiones

El Workflow Manager ofrece la posibilidad de añadir expresiones que actúan como condiciones a cumplir. Estas expresiones se pueden incluir en los enlaces entre tareas, una Decision o una tarea de asignación.

En este editor de expresiones podemos ver las siguientes funciones:

- Funciones SQL diseñadas para tratar expresiones corrientes.
- Funciones predeterminadas para la transición entre tareas.
- Funciones basadas en parámetros de Powercenter.

Después de definir la expresión en el editor, es necesario validarla haciendo clic en Validate. No es posible ejecutar un workflow que contiene una expresión no válida. El Workflow Manager invalida el workflow también.

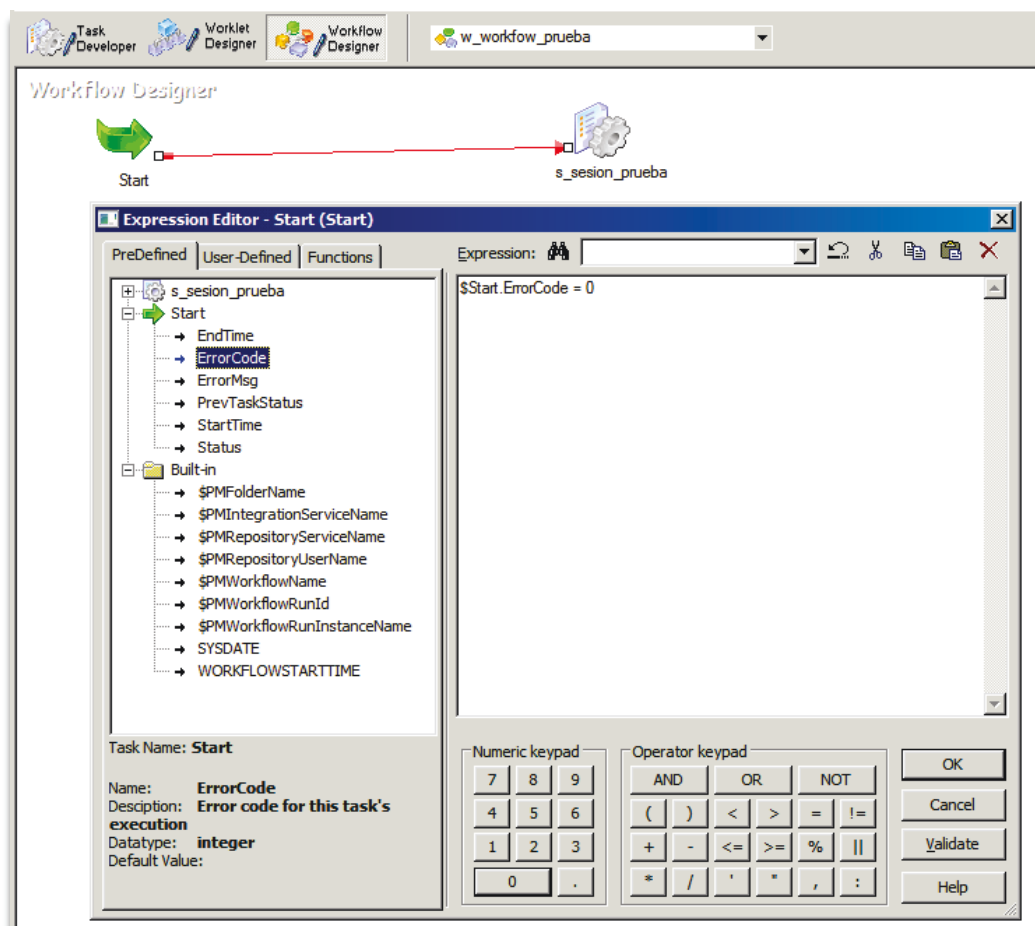


Figura 11: Editor de expresiones

3.11. Conexiones

Antes de crear el workflow hay que crear una conexión con la base de datos que se va a utilizar. Algunos tipos de conexiones también cuentan con un subtipo de conexiones. Por ejemplo, un tipo de conexión relacional tiene subtipos como por ejemplo Oracle y Microsoft SQL Server. Define los valores para la conexión basándose en el tipo de conexión y subtipo. Por ejemplo, un tipo de conexión relacional tiene subtipos como por ejemplo Oracle y Microsoft SQL Server.

Al crear una sesión, se tiene que asignar un tipo de conexión para cada tabla. Las conexiones y esquemas pueden ser diferentes para cada tabla de las que puede tener una sola sesión.

En la siguiente tabla se describen los tipos de conexión que se pueden crear o elegir al configurar una sesión:

Conexiones	Descripción
Relational	Conexión relacional de origen, de destino, de búsqueda, o base de datos de procedimiento almacenado. Al configurar una sesión, no se puede cambiar el tipo de conexión relacional.
FTP	Conexión FTP o SFTP. Al configurar una sesión, se elige un tipo de conexión FTP para acceder a los archivos planos o archivos XML a través de FTP. Hay que especificarlo al configurar opciones de origen o de destino.
Loader	Conexión relacional para el cargador externo de datos para el destino, como IBM DB2 Autoloader o Teradata FastLoad. Al configurar una sesión, seleccionamos File como el tipo de escritura para la instancia de destino relacional. Seleccionamos una conexión Loader para cargar los archivos de salida de Teradata, Oracle, DB2 o Sybase IQ a través de un cargador externo.
Queue	Seleccione un tipo de conexión de Queue para acceder a una fuente MSMQ o WebSphere MQ.

Application	<p>Conexión que establecemos con la aplicación de origen o destino, como Netezza o SAP NetWeaver.</p> <p>Se utiliza para acceder a fuentes y destinos de PowerExchange y fuentes Teradata FastExport. También se puede acceder a transformaciones como HTTP, Salesforce Lookup y BAPI / RFC.</p>
None	<p>Tipo de conexión no disponible en el Explorador de conexión.</p> <p>Al configurar una sesión, se selecciona <i>None</i> si el mapping contiene un archivo plano o XML como fuente o destino.</p>

Para añadir una conexión: **Connections /Relational...**

Tenemos las siguientes figuras:

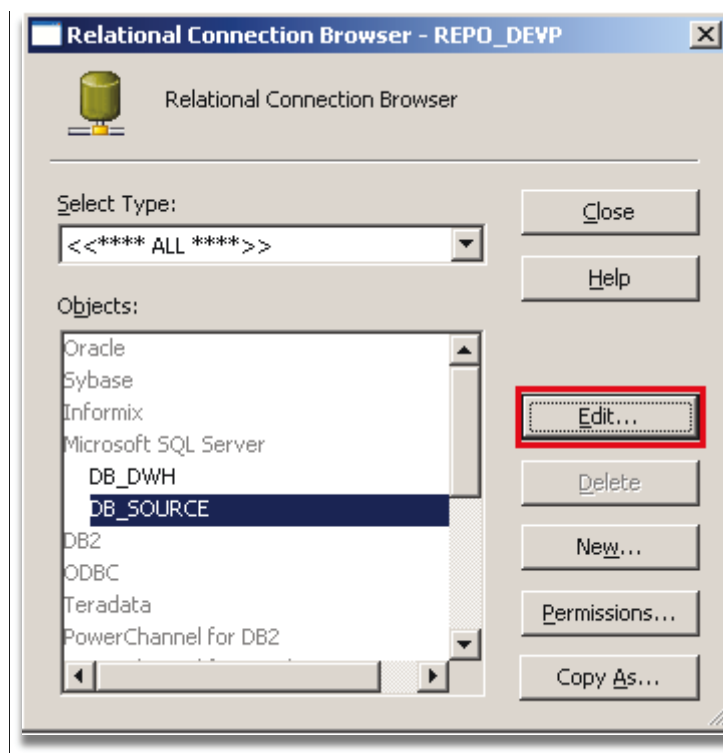


Figura 12: Configurar conexión

Se elige el esquema deseado y a continuación se introduce la información necesaria en siguiente página.

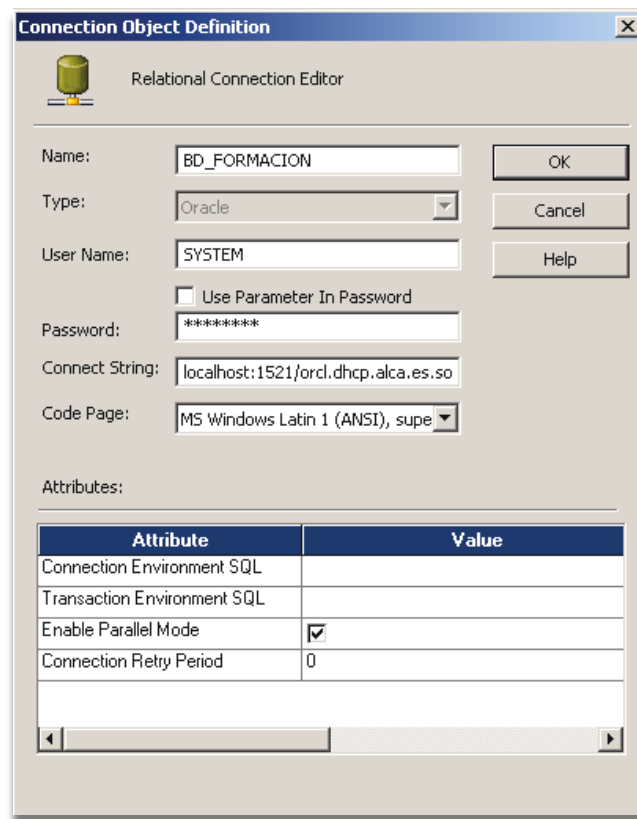


Figura 13: Conexión con BBDD

3.12. Workflow y Worklet

3.12.1. Definición del workflow

Un workflow es un conjunto de instrucciones que indican al Servicio de Integración cómo ejecutar tareas como pueden ser, sesiones, notificaciones de correo o líneas de comandos Shell. Después de crear las tareas deseadas, se integran en el workflow conectándose entre ellas a través de enlaces configurables. En el Workflow Designer se puede especificar condiciones en los enlaces, utilizar variables y parámetros así como tareas de tipo Evento que controlan la secuencia de ejecución del conjunto de tareas que contiene el workflow.

Muchas veces se necesita repetir un conjunto de tareas, de manera común, dentro de algunos workflows. El Workflow Manager ofrece la posibilidad de reutilizar estas tareas de manera automatizada, por así decirlo, haciendo uso de los worklets. Un worklet es muy similar a un workflow, con la diferencia de que el primero se puede integrar en uno o varios workflows. Esto nos permite reutilizar unas tareas comunes en varios workflows diferentes sin tener que volver a crearlas en cada uno de ellos.

a. Crear un workflow

La creación de un Workflow se puede resumir en la secuencia siguiente:

- Crear el workflow: En el espacio de trabajo Workflow Designer, hacer clic en la pestaña **Workflow > Create** y añadir un nombre. Generalmente se suele poner la letra 'w_' como prefijo y un nombre significativo indicando la función del workflow. El workflow se crea con la tarea **Start** por defecto y no se puede borrarla.

Es muy importante añadir una descripción que explica la función del workflow.

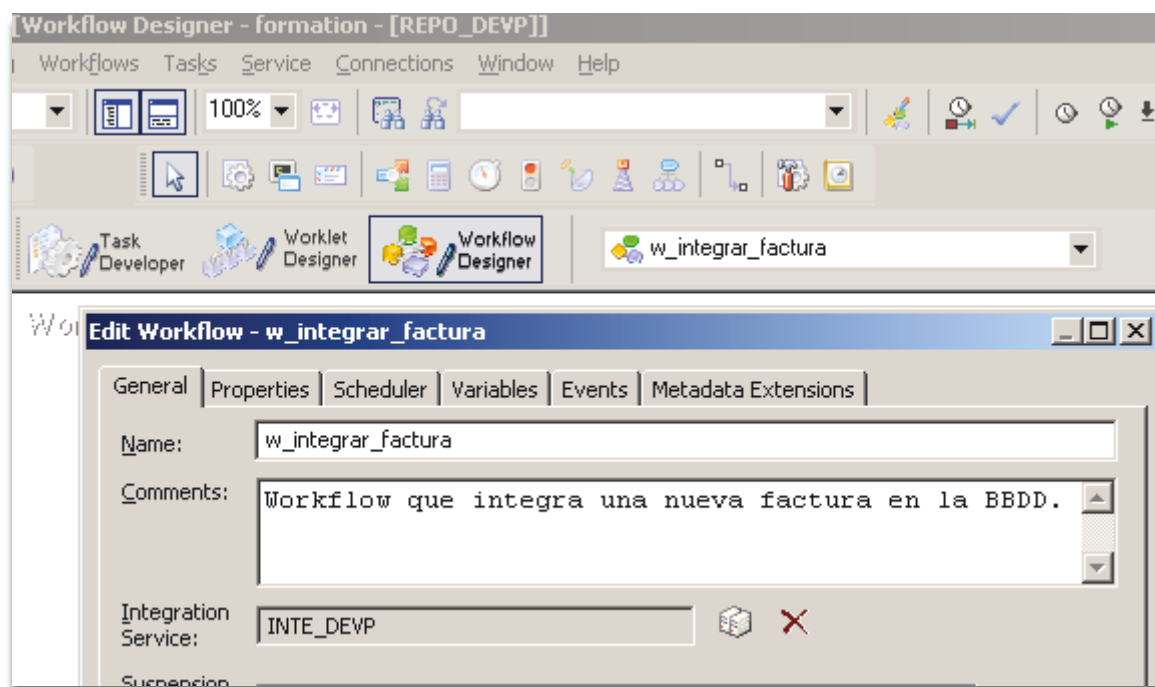


Figura 14: Crear un workflow

- Añadir una tarea arrastrándola desde las carpetas en el navegador **Repository Navigator**. Estas tareas pueden ser de tipo tarea, sesión o worklet.

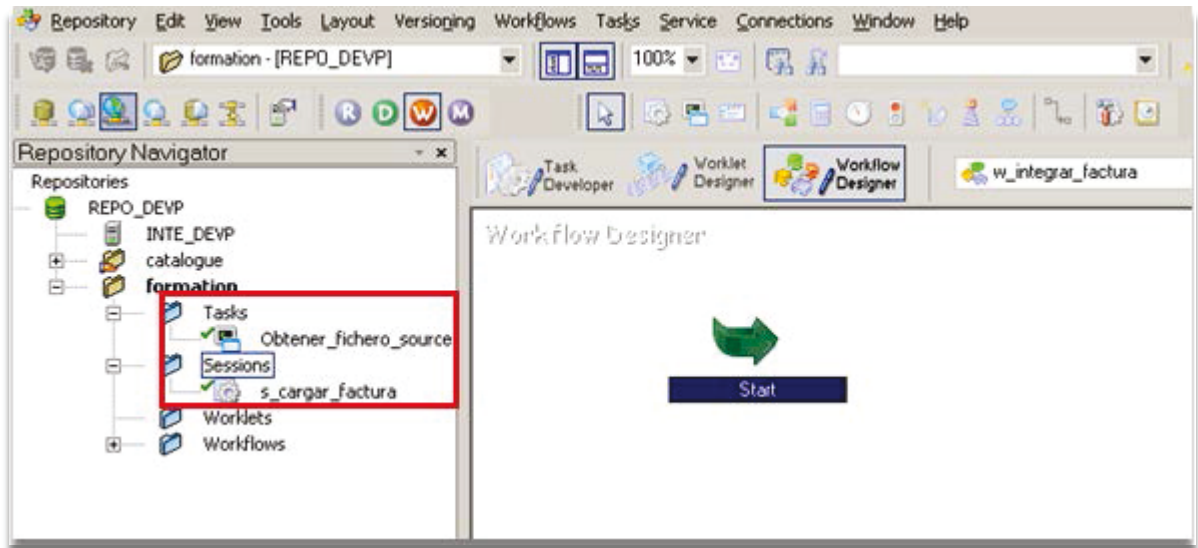


Figura 15: Añadir sesiones creadas en el workflow

- Para crear una sesión o tarea inexistente desde el mismo workflow, hacer clic en el icono correspondiente en la barra de menú y hacer clic dentro del espacio de trabajo.



Figura 16: Tipos de tareas

Otra forma para crear una sesión es: **Tasks/Create...**

Seleccionar el tipo de tarea que queremos crear y añadir un nombre significativo. Las sesiones suelen tener una nomenclatura con el prefijo 's_'. Clic en **Create**.

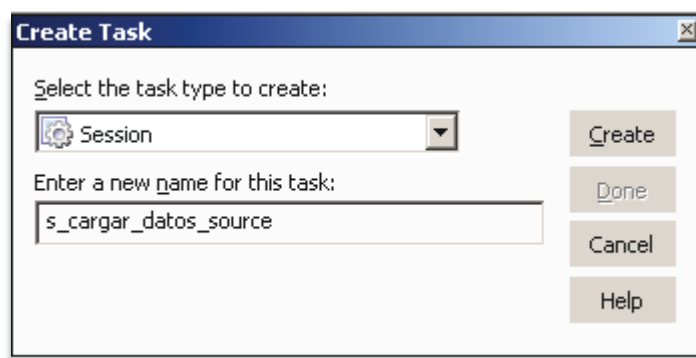


Figura 17: Creación de una sesión o tarea

Seleccionar el mapping para el cual se desea crear la sesión en la siguiente pantalla:

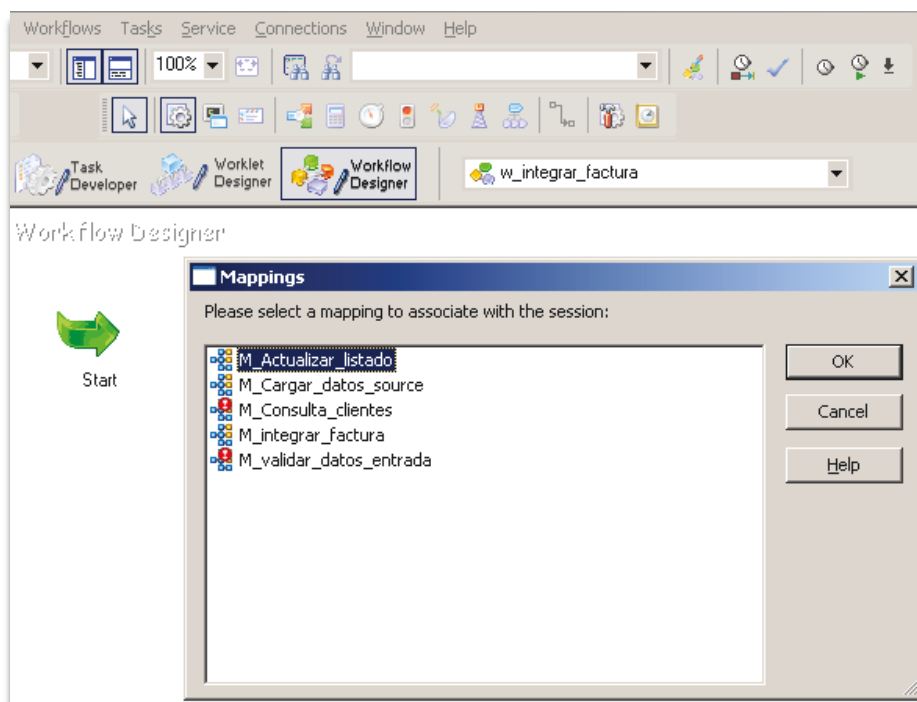



Figura 18: Elección del Mapping relacionado con la sesión

Nota:

Nótese que es necesario que el mapping este valido y no tenga el símbolo rojo al lado.

- Una vez creada la sesión, hay que conectarla con la tarea **Start** con un enlace haciendo clic en el símbolo  y conectando las dos tareas como aparece en la siguiente pantalla:

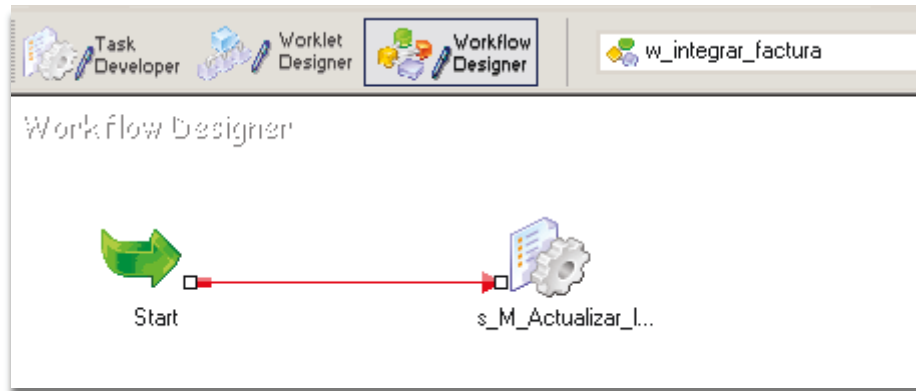



Figura 19: Enlazar una sesión

- Validar el workflow y guardarlo haciendo **Control+s**
- Ejecutar el workflow haciendo clic sobre el icono 

3.12.2. Configuración del Workflow

a. Pestaña General

Para editar el workflow: **Workflow/Edit..**

Es muy importante no olvidar la descripción (sin acentos ni comillas) funcional del workflow ya que, por ejemplo, en un proyecto de mantenimiento evolutivo, los desarrolladores tienen que tener una idea sobre el funcionamiento del workflow sobre el cual van a realizar cambios. Este detalle puede llegar a facilitar mucho el trabajo.

Opción	Descripción
Name	Nombre del workflow.
Comments	Descripción funcional del workflow.

Integration Service	Indicar aquí el motor de Powercenter (Servicio de integración) que va a ejecutar el workflow.
Suspension Email	Mensaje que el SI envía por correo cuando falla la tarea y se suspende la ejecución del workflow.
Disabled	Desactiva el workflow de la programación Scheduler. El SI no ejecuta el workflow mientras este desactivado.
Suspend on Error	El SI suspende el workflow cuando detecta una tarea fallida en el mismo.

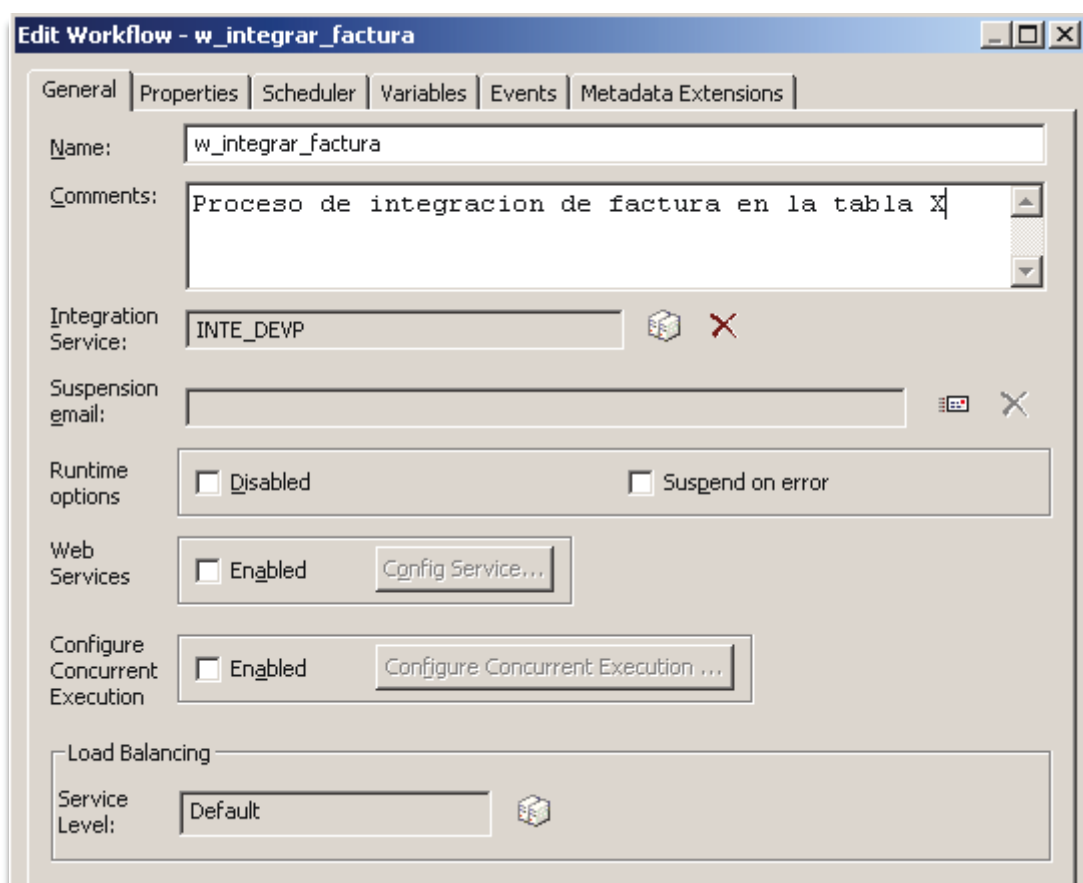


Figura 20: Opciones generales

b. Pestaña Properties

La siguiente figura muestra la pestaña *Properties* del Workflow:

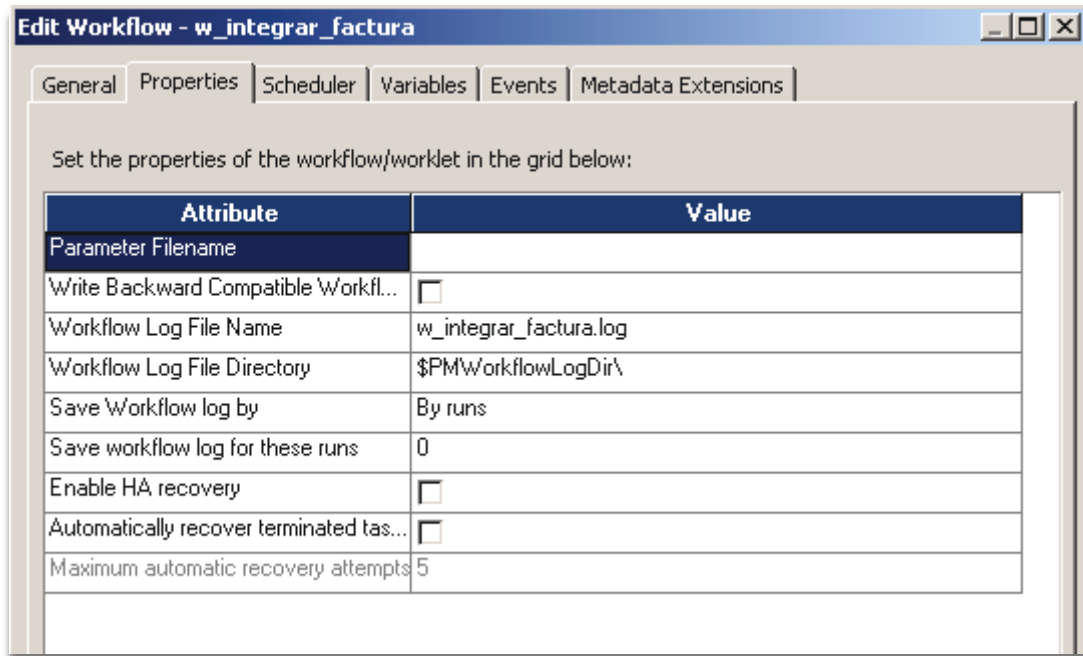


Figura 21: Pestaña *Properties* de edición del Workflow

Opciones:

Opción	Descripción
Parameter Filename	Directorio y nombre del fichero de parámetros del Workflow
Write Backward Compatible Workflow	Marcar si se quiere obtener un log de la ejecución del Workflow
Workflow Log File Name	Nombre del Workflow
Workflow Log File Directory	Directorio \$PMWorkflowLogDir\
Save workflow log by	Opciones By runs By timestamp
Save workflow log for these runs	Número de logs históricos que guarda. 0 (0 más son guardados. Solo el actual es guardado)

c. Pestaña Scheduler

El Workflow Manager proporciona la posibilidad de programar el lanzamiento de workflows de manera periódica según el criterio que elijamos. Cada workflow tiene un programador asociado. Un programador es un objeto de repositorio que contiene un conjunto de parámetros de planificación.

Se puede crear un programador para un único uso o que sea reutilizable para utilizar el mismo conjunto de parámetros de planificación para los workflows de la misma carpeta.

Algunos detalles a tener en cuenta y buenas prácticas:

- Si se elimina el programador asociado a un workflow, el servicio de integración marca el workflow como inválido y hay que volver a validarlo antes de ejecutarlo.
- Si se elimina una carpeta, el Servicio de integración elimina todas las programaciones. Al copiar una carpeta, el SI reprograma automáticamente todos los workflows de la misma.

El SI no ejecuta el workflow en los siguientes casos:

- Fallo de una ejecución previa. Cuando un workflow falla el SI elimina la programación del mismo y hay que rehacerlo manualmente.
- Cuando el proceso del SI falla por alguna razón ajena al propio workflow. También hace falta reprogramarlo de nuevo.
- Cuando el Servicio de integración funciona en modo seguro.

La siguiente figura muestra la pestaña Scheduler del Workflow:

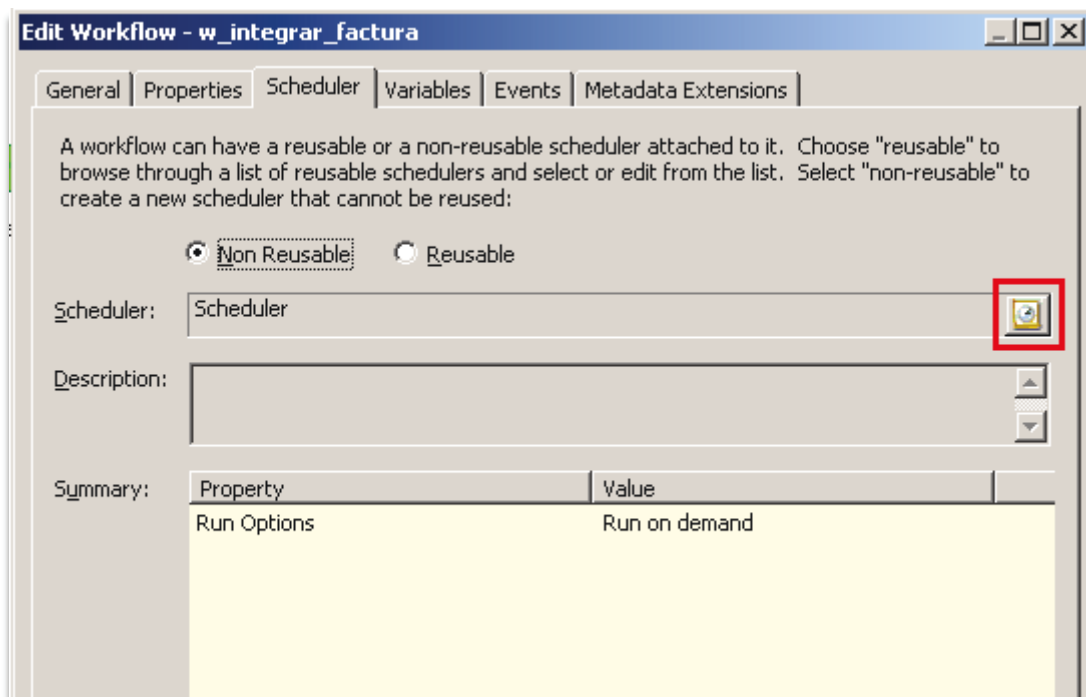


Figura 22: Pestaña Scheduler de edición del Workflow

En esta pestaña podemos programar la hora y días de ejecución:

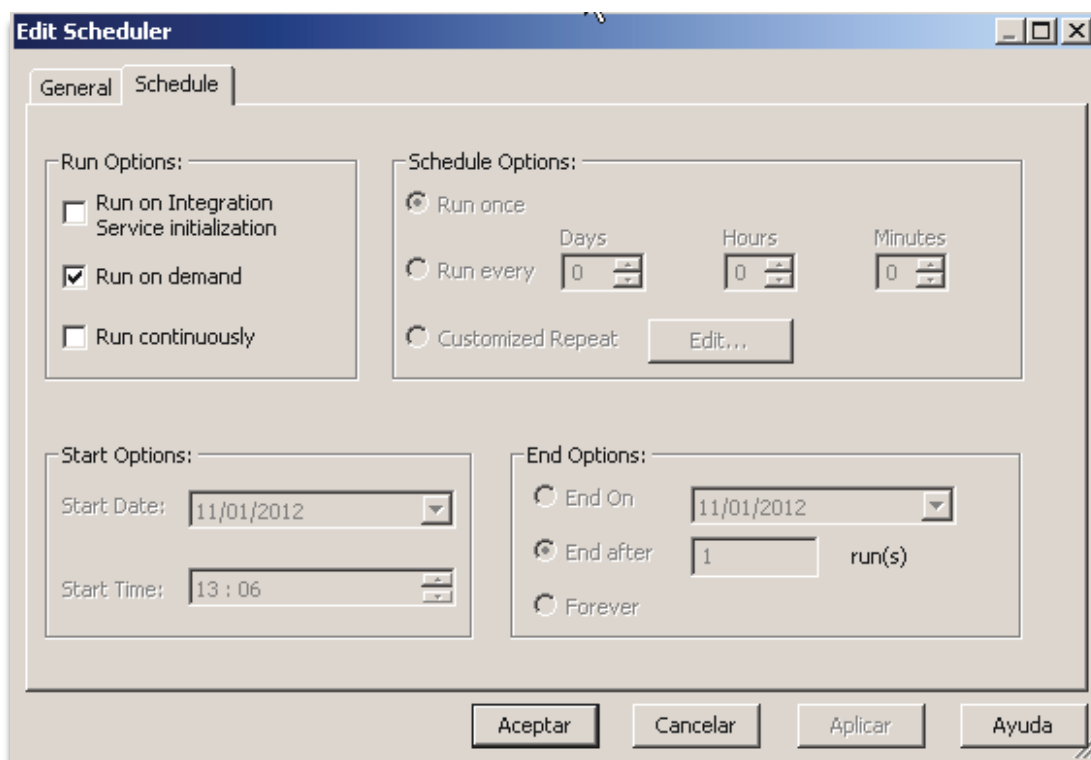


Figura 23: Scheduler

d. Pestaña Variables

Aquí se declaran las variables que va a utilizar el workflow. Se puede crear una variable de cualquier tipo y hacer que sea persistente o no. Persistente es cuando el SI hace que la variable mantenga el mismo valor que tenía en las ejecuciones precedentes.

La siguiente figura muestra la pestaña Variables del Workflow:

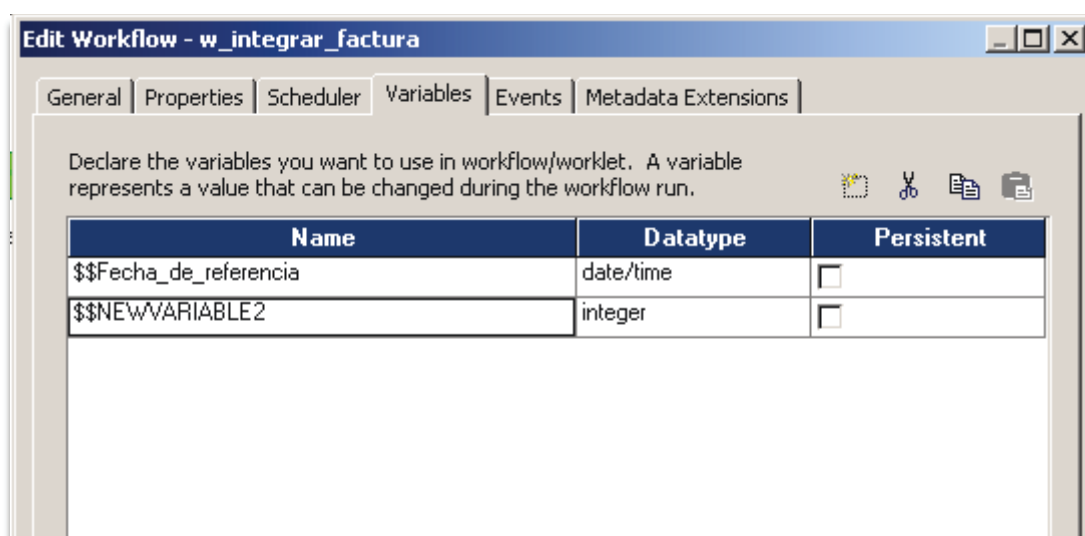


Figura 24: Pestaña Variables

e. Tarea de envío de correo

Consiste en enviar un correo electrónico a los destinatarios designados al ejecutarse dentro de un workflow. Para ello, hace falta crear primero la tarea en el gestor de tareas y posteriormente incluirla en el workflow.

Esta tarea es muy útil en los casos de que los gestores de una aplicación quieran realizar un control instantáneo sobre alguna ejecución. Por ejemplo, en una aplicación de carácter crítico en cuanto a disponibilidad, donde se realizan operaciones de ETL que se suelen hacer por la noche o durante un fin de semana por ejemplo, es muy

importante seguir de cerca el transcurso de la carga. Este tipo de tareas se incluyen en los workflows potencialmente para informar de interrupciones o fallos de ejecución (o fin de carga sin errores) y de esta manera poder intervenir a tiempo para desbloquear la situación.

f. Validación del workflow

Antes de proceder a ejecutar el workflow se debe validarlo. Cuando lo validamos, el SI valida todas las instancias de tareas que contiene, incluyendo los worklets.

El SI también valida las estructuras de la fuente y el destino (Tablas o ficheros). Si alguna estructura sufre algún cambio, el workflow aparece como impactado. Un error muy común entre los desarrolladores es cuando se modifica alguna estructura sin actualizar el workflow. El WM permite lanzar el workflow pero con la configuración anterior al cambio y puede fallar la ejecución.

3.12.3. Definición de la Sesión

Una sesión es el conjunto de instrucciones que indican al servicio de integración cómo y cuándo mover los datos de una fuente a un destino. Una sesión es un tipo de tarea similar a otras tareas disponibles en el gestor de workflows.

Se puede crear una sesión en el editor de sesiones o directamente dentro el mismo workflow que la va a contener y tiene que estar construida sobre un mapping válido y sin errores. Las sesiones solo pueden ejecutarse dentro del workflow que las contiene y después de haber configurado las opciones que veremos en el punto siguiente.

Ejecución de sesiones:

- Se puede ejecutar una o varias sesiones dentro de un worflow sin ejecutar la totalidad de sesiones.
- Ejecutar una sesión: Clic derecho sobre la sesión > **Start Task**.
- Ejecutar una secuencia de sesiones a partir de una de ellas.

3.12.4. Configuración de la Sesión

Una vez creada la sesión, procedemos a su configuración como veremos en los ejemplos a continuación.

a. Pestaña General

La siguiente figura muestra la pestaña **General** de la sesión:

The screenshot shows the 'Edit Tasks' dialog box with the 'General' tab selected. The dialog has a title bar with standard window controls. Below the title bar is a tabbed interface with tabs for 'General', 'Properties', 'Config Object', 'Mapping', 'Components', and 'Metadata Extensions'. The 'General' tab is active. It contains the following fields and controls:

- Select task:** A dropdown menu showing 's_M_Actualizar_listado' with a 'Rename' button to its right.
- Task type:** A dropdown menu showing 'Session' and a 'Make reusable' checkbox.
- Description:** A text area containing 'Sesion que actualiza la tabla X'.
- Mapping name:** A text field containing 'M_Actualizar_listado'.
- Resources:** A text field and an 'Edit' button.
- Fail parent if this task fails:** A checked checkbox.
- Fail parent if this task does not run:** An unchecked checkbox.
- Disable this task:** An unchecked checkbox.
- Treat the input links as:** A group box containing two radio buttons: 'AND' (selected) and 'OR'.

At the bottom of the dialog are four buttons: 'OK', 'Annuler', 'Appliquer', and 'Aide'.

Figura 25: Pestaña General

Normas y buenas prácticas:


Opción	Descripción
Fail parent if this task fails	El padre (workflow) aparecerá como <i>Failed</i> cuando se ejecute si esta tarea falla. (Se utiliza casi por defecto)
Fail parent if this task does not run	El padre (workflow) aparecerá como <i>Failed</i> cuando se ejecute si esta tarea no se ejecuta. (Sólo si funcionalmente es necesario)
Descripción	Descripción funcional del proceso. Es recomendable que no contenga ni acentos ni comillas.
Disable this task	Desactiva la tarea. (Sólo si es necesario)
Treat the Input Links as AND or OR	Ejecuta la tarea cuando una o todas las condiciones del enlace de entrada (que la une con la(s) sesión(es) anterior(es)) están a True. AND: se tienen que cumplir todas. OR: al menos una de las condiciones.

b. Pestaña Properties

La siguiente figura muestra la pestaña **Properties** de la sesión:

Edit Tasks

General Properties Config Object Mapping Components Metadata Extensions

Select task:  s_M_Actualizar_listado

Task type: Session

Attribute	Value
General Options	
Write Backward Compatible Session Log File	<input type="checkbox"/>
Session Log File Name	s_M_Actualizar_listado.log
Session Log File directory	\$PMSessionLogDir\
Parameter Filename	
Enable Test Load	<input type="checkbox"/>
Number of rows to test	1
\$Source connection value	Relational:DB_SOURCE
\$Target connection value	Relational:DB_DWH
Treat source rows as	Data driven
Commit Type	Target
Commit Interval	10000
Commit On End Of File	<input checked="" type="checkbox"/>
Rollback Transactions on Errors	<input type="checkbox"/>
Recovery Strategy	Fail task and continue workflow
Java Classpath	
Performance	
DTM buffer size	Auto
Collect performance data	<input type="checkbox"/>
Write performance data to repository	<input type="checkbox"/>
Incremental Aggregation	<input type="checkbox"/>
Reinitialize aggregate cache	<input type="checkbox"/>
Enable high precision	<input type="checkbox"/>
Session retry on deadlock	<input type="checkbox"/>
Pushdown Optimization	None
Allow Temporary View for Pushdown	<input type="checkbox"/>
Allow Temporary Sequence for Pushdown	<input type="checkbox"/>
Allow Pushdown for User Incompatible Conne...	<input type="checkbox"/>
Session Sort Order	Binary

General Options

General

OK Annuler Appliquer Aide

Figura 26: Pestaña Properties

Normas y buenas prácticas:

General Options:

Detallaremos las opciones que más se utilizan.

Opción	Descripción
Write Backward Compatible Session Log File	Si la opción está marcada crea un fichero log de la sesión.
Session Log File Name	El nombre del log la sesión.
Session Log File Directory	Directorio donde se creará el log.
Parameter Filename	Directorio y nombre del fichero de parámetros de la sesión. <i>\$PMRootDir\Parameter\fichero_parametros.txt</i>
Enable Test Load	Se puede realizar una carga de datos de prueba. El IS lee los datos, los transforma pero no los carga en la tabla destino. Ejecuta la sesión completa con toda la configuración pero no llega a cargar nada en el destino. (No se suele utilizar)
Number of rows to test	Número de registros que se desean cargar como prueba.
\$Source connection value	Conexión del esquema donde reside la fuente. (Se suele indicar en la pestaña Mapping y deja aquí vacío)
\$Target connection value	Ídem.
Treat source rows as	Indica al IS cómo tratar los datos del Source : Insert, Update, Delete o Data Driven (cuando tenemos transformaciones Update Strategy). Si seleccionamos Data Driven, el tipo de carga tiene que ser Normal , sino la sesión falla.
Commit Type	Determina si el SI usa un commit en Target, en Source o User-defined. Si el Mapping no tiene la transformación Transaction Control , se puede elegir entre Target o Source. Por defecto, Target. Si el Mapping tiene la transformación Transaction Control: por defecto User-defined
Commit Interval	Por defecto hace commit cada 10.000 filas.
Recovery Strategy	Por defecto <i>Fail task and continue workflow</i>

Para las opciones relacionadas con el **Performance** ver la ayuda de Powercenter. Las opciones que más se utilizan son las siguientes:

Opción	Descripción
DTM buffer size	Por defecto se deja en Auto, salvo que se indique el tamaño en función de la volumetría de tablas.
Enable high precision	Un error muy común es cuando leemos datos de tipo numérico con precisión mayor a 18. Hay que indicar al IS esta opción para saber tratar cifras con mayor precisión.

c. Pestaña Config Object

La siguiente figura muestra las opciones que se pueden configurar aquí. Citaremos las más utilizadas:

Edit Tasks

General | Properties | **Config Object** | Mapping | Components | Metadata Extensions

Select task: **S_M_Actualizar_listado**

Task type: Session

Config Name: default_session_config

Attribute	Value
Advanced	
Constraint based load ordering	<input type="checkbox"/>
Cache LOOKUP() function	<input checked="" type="checkbox"/>
Default buffer block size	Auto
Line Sequential buffer length	1024
Maximum Memory Allowed For Auto Memory Attr...	512MB
Maximum Percentage of Total Memory Allowed ...	5
Additional Concurrent Pipelines for Lookup Cac...	Auto
Custom Properties	
Pre-build lookup cache	Auto
DateTime Format String	MM/DD/YYYY HH24:MI:SS.US
Pre 85 Timestamp Compatibility	<input type="checkbox"/>
Log Options	
Save session log by	Session runs
Save session log for these runs	0
Error handling	
Stop on errors	0
Override tracing	None
On Stored Procedure error	Stop
On Pre-session command task error	Stop
On Pre-Post SQL error	Stop
Error Log Type	None
Error Log DB Connection	
Error Log Table Name Prefix	
Error Log File Directory	\$PMBadFileDir\
Error Log File Name	PMEError.log
Log Row Data	<input checked="" type="checkbox"/>
Log Source Row Data	<input type="checkbox"/>
Data Column Delimiter	
Partitioning Options	
Dynamic Partitioning	Disabled
Number of Partitions	1
Session on Grid	
Is Enabled	<input type="checkbox"/>
Advanced	
Advanced options group for session configuration	

OK Annuler Appliquer Aide

Figura 27: Pestaña Config Object

Normas y buenas prácticas:

- Advanced:

Opción	Descripción
Cache Lookup() Function	Si está marcada, el SI almacena en caché funciones Lookup PowerMart 3.5 en el Mapping. Marcada por defecto
Default Buffer Block Size	Tamaño de los bloques buffer usados para mover cachés de datos e índices de Sources a targets. Por defecto, el SI determina este valor en tiempo de ejecución. Por defecto Auto
Line Sequential Buffer Length	Incrementar el valor por defecto (1024 bytes por línea) si las líneas de los ficheros de entrada son mayores de 1024 bytes. Por defecto 1024
Maximum Memory Allowed for Auto Memory Attributes	Máxima memoria destinada a caché automático cuando se configura el SI para determinar el tamaño del caché de la sesión en tiempo de ejecución. Defecto 512 MB
Maximum Percentage of Total Memory Allowed for Auto Memory Attributes	Máximo porcentaje de memoria total destinada a caché automático cuando se configura el SI para determinar el tamaño del caché de la sesión en tiempo de ejecución. Defecto 5
Additional Concurrent Pipelines for Lookup Cache Creation	Por defecto Auto
Pre-built Lookup cache	Por defecto Auto
DateTime Format String	Formato de la cadena de fechas. Opciones: MM/DD/YYYY HH24:MI:SS, especifica segundos. MM/DD/YYYY HH24:MI:SS.MS, especifica milisegundos. MM/DD/YYYY HH24:MI:SS.US, especifica microsegundos. MM/DD/YYYY HH24:MI:SS.NS, especifica nanosegundos. Para Teradata : MM/DD/YYYY HH24:MI:SS.US

- Log Options:

Opción	Descripción
Save Session Log By	Opciones: Session Timestamp: El Log Manager salva todos los logs de las sesiones, añadiendo un timestamp a cada log creado. Session Runs: El Log Manager salga un número de logs definido en la opción <i>Save Session Log for These Runs</i> .
Save Session Log for These Runs	Número de logs históricos que se salvan.

- Error Handling:

Opción	Descripción
Stop On Errors	Indica cuantos errores 'non-fatal' que el SI puede encontrar antes de que pare la sesión.
On Stored Procedure Error	Requerido si la sesión usa procedimientos almacenados.
On Pre-Session Command Task Error	Requerido si la sesión usa comandos pre-sesión
On Pre-Post SQL Error	Requerido si la sesión usa SQL

Para las opciones relacionadas con el **Particionado** ver la ayuda de Powercenter.

d. Pestaña Mapping

La siguiente figura muestra la pestaña Mapping de la sesión:

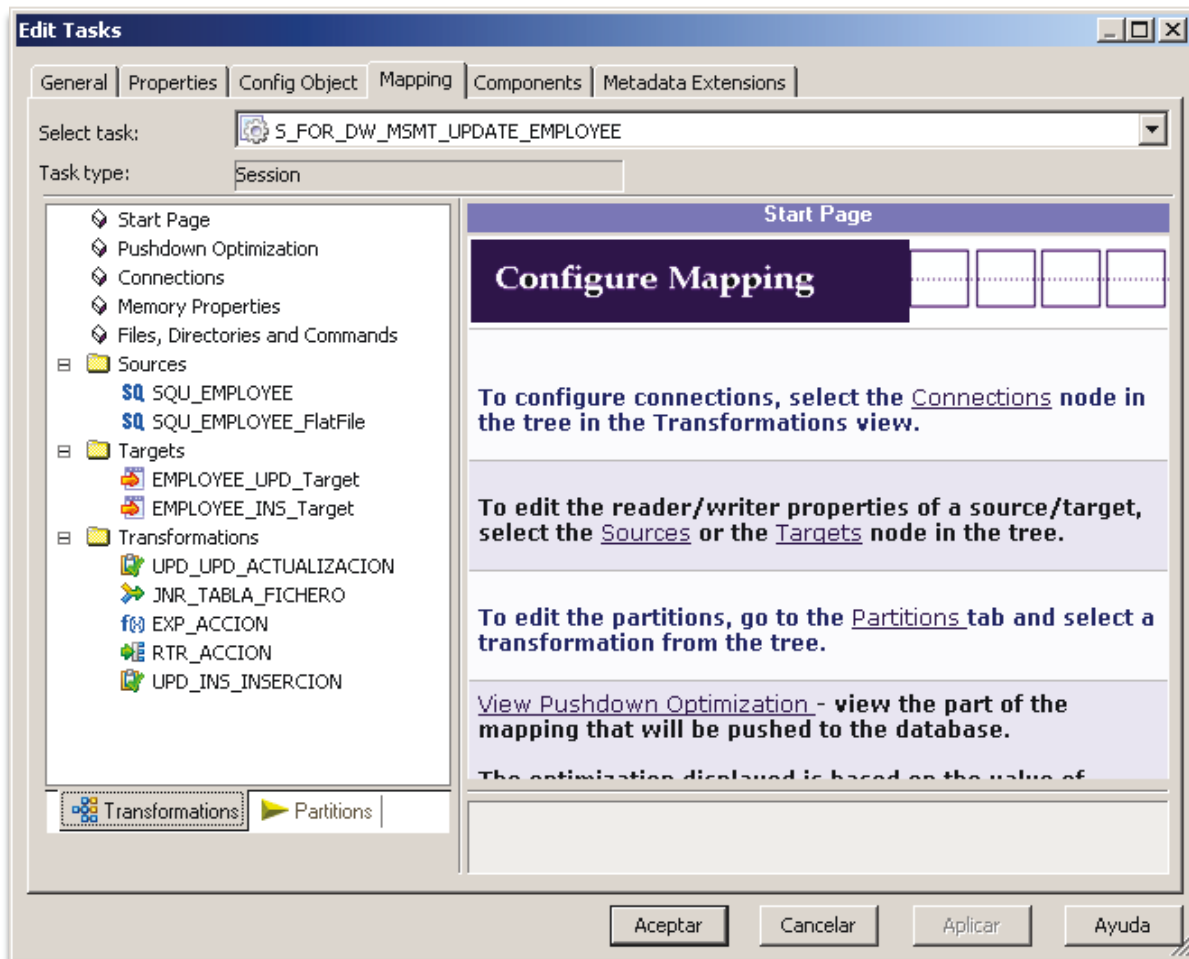


Figura 28: Pestaña Mapping

- Connections

Clicando sobre el apartado *Connections* seleccionaremos las conexiones para cada objeto.

Podemos parametrizar las conexiones a los parámetros \$Source y \$Target. De esta forma, en vez de elegir la conexión para cada objeto, podemos parametrizarla y luego elegir el parámetro para cada objeto.

En ocasiones solemos tener varias tablas de diferentes esquemas, por lo que tenemos que asignar la conexión correspondiente manualmente.

En el ejemplo que hemos elegido aquí, utilizamos los parámetros y varias estructuras diferentes en Source y Target para ver las diferencias.

Nota:

Cada proyecto determinará la forma de asignar las conexiones.

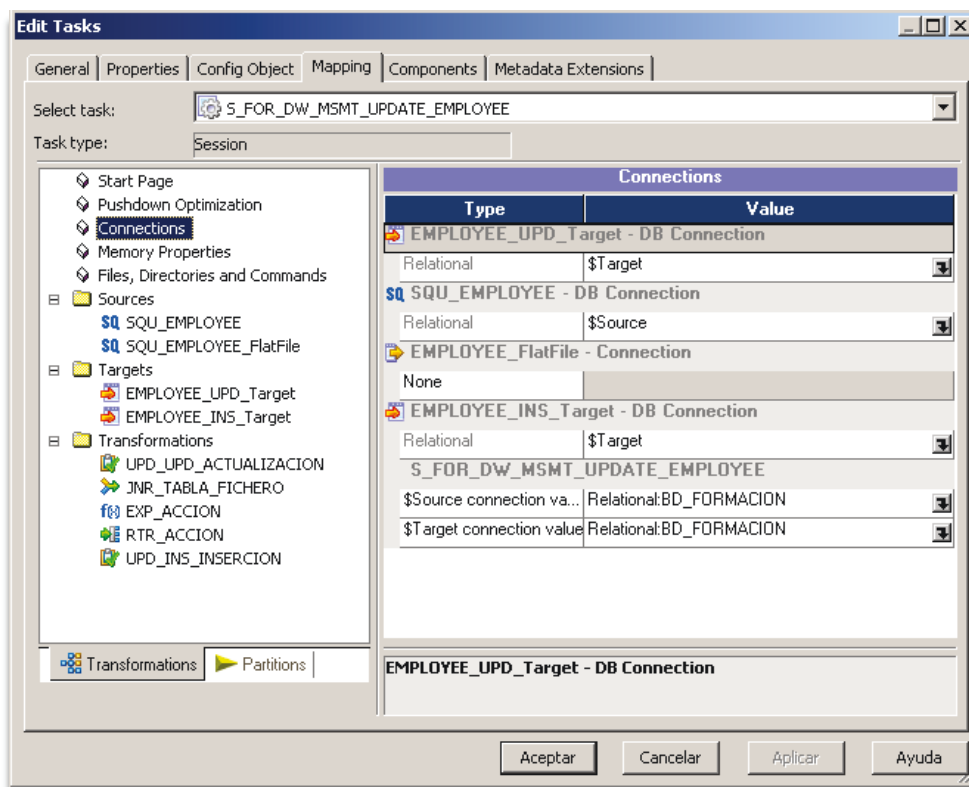


Figura 29: Pestaña Mapping - Connections

➤ Sources

Caso 1:

Cuando la fuente es una tabla.

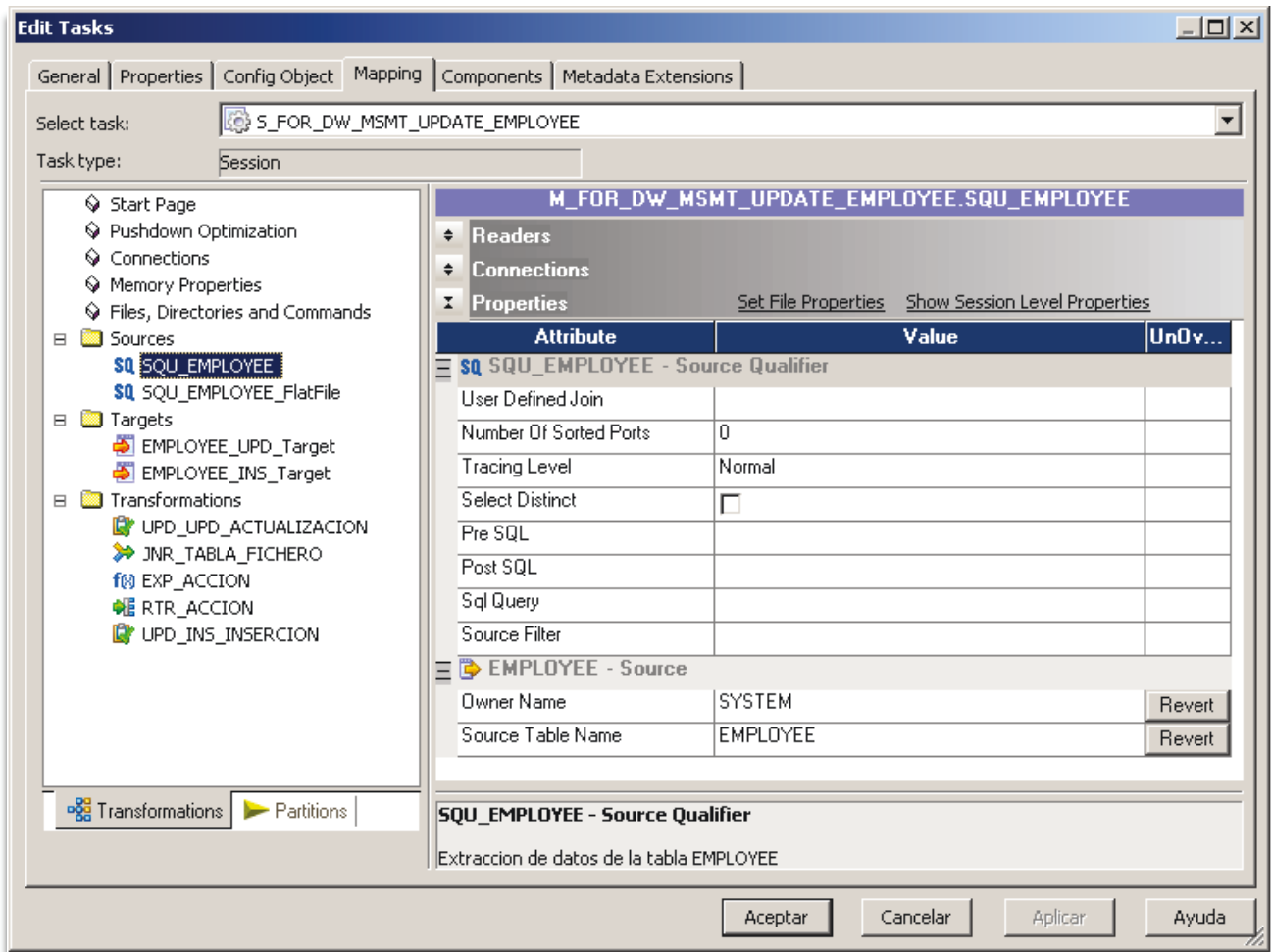


Figura 30: Pestaña Mappings – Sources Tabla

Opciones:

Opción	Descripción
Owner Name	Esquema de la BBDD que contiene la tabla
Source Table Name	Nombre de la Tabla

Caso 2:

Cuando la fuente es una tabla.

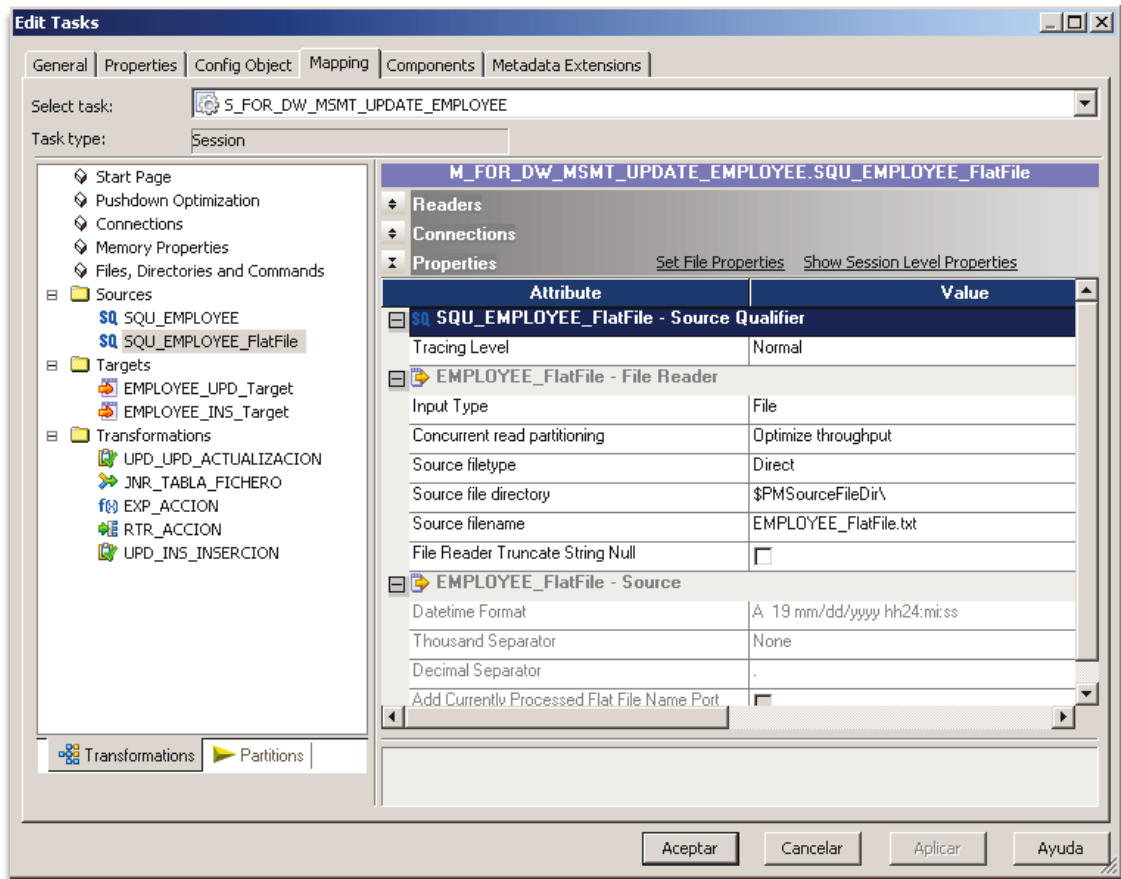


Figura 31: Pestaña Mappings – Sources fichero

Opciones:

Opción	Descripción
Source filetype	Direct / Indirect La opción Direct lee directamente el fichero especificado
Source file directory	Directorio donde se ubica el fichero
Source filename	Nombre del fichero

Clicando en la opción *Set File Properties*, definimos las propiedades de fichero, si los campos del fichero tienen un ancho fijo o existe un delimitador de campos.

Esto es bastante importante a la hora de realizar pruebas, ejecutando workflows y consultando el fichero de salida, de esta manera podemos identificar los campos con la ayuda de los separadores.

Finalmente, el cliente es el que decide el formato que tiene que tener el fichero de salida.

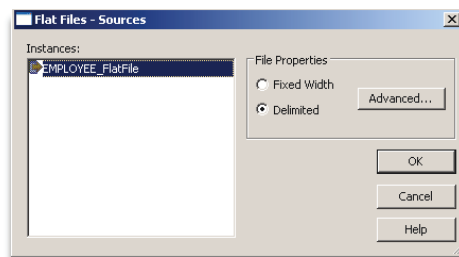


Figura 32: Set File Properties

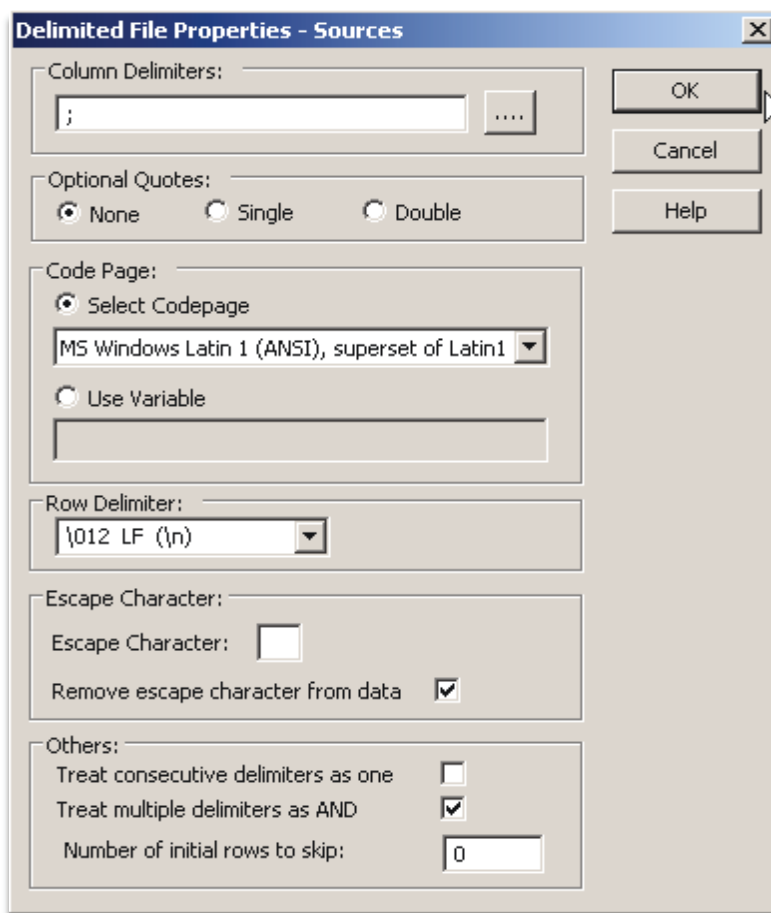


Figura 33: Set File Properties – Delimited File

➤ Targets

Caso 1:

Cuando el destino es una tabla.

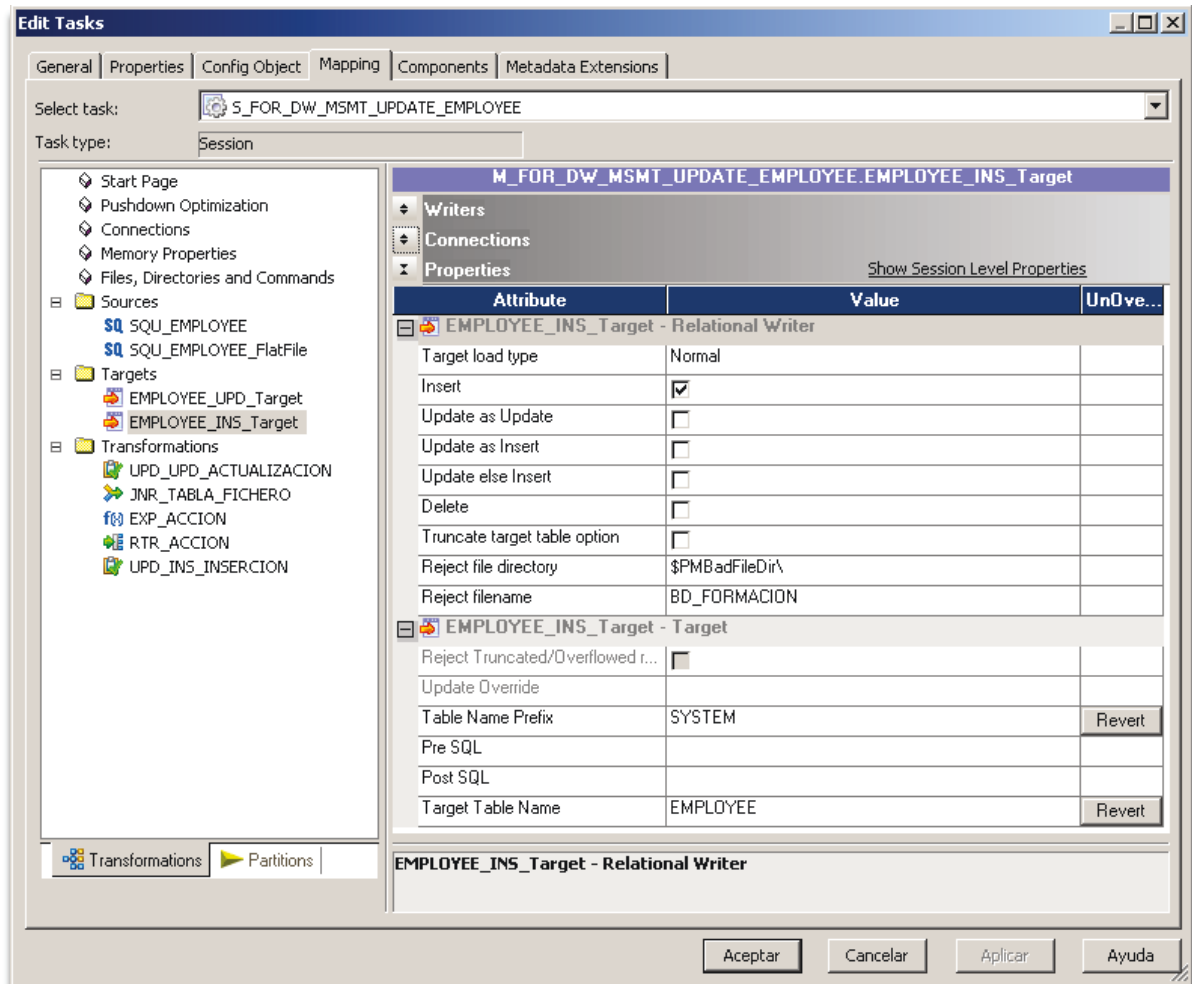


Figura 34: Pestaña Mappings - Targets

Se deben completar las opciones:

Opción	Descripción
Target Load type	<p>Se puede elegir Normal o Bulk.</p> <p>Normal: El SI carga el Target normalmente.</p> <p>Elegir si el Mapping contiene Update Strategy.</p> <p>Bulk: Solo con DB2, Sybase, Oracle o Microsoft SQL Server.</p> <p>Puede mejorar el rendimiento pero limita la posibilidad de</p>

	recuperación ya que no hay logs. La tabla no debe estar indexada.
Insert	El SI inserta todas las filas marcadas como insert
Update as Update	El SI actualiza todas las filas marcadas como update
Update as Insert	El SI inserta todas las filas marcadas como update
Update else Insert	El SI actualiza las filas marcadas como update si existen en el target, e inserta las demás marcadas como insert.
Delete	El SI elimina todas las filas marcadas como delete.
Truncate Table	El SI trunca la tabla target antes de cargar.
Reject file directory	Por defecto <i>\$PMBadFileDir</i> . Añadir el proyecto.
Table Name Prefix	Esquema de la BBDD que contiene la tabla
Target Table Name	Nombre de la Tabla

Caso 2:

Cuando el destino es un fichero plano.

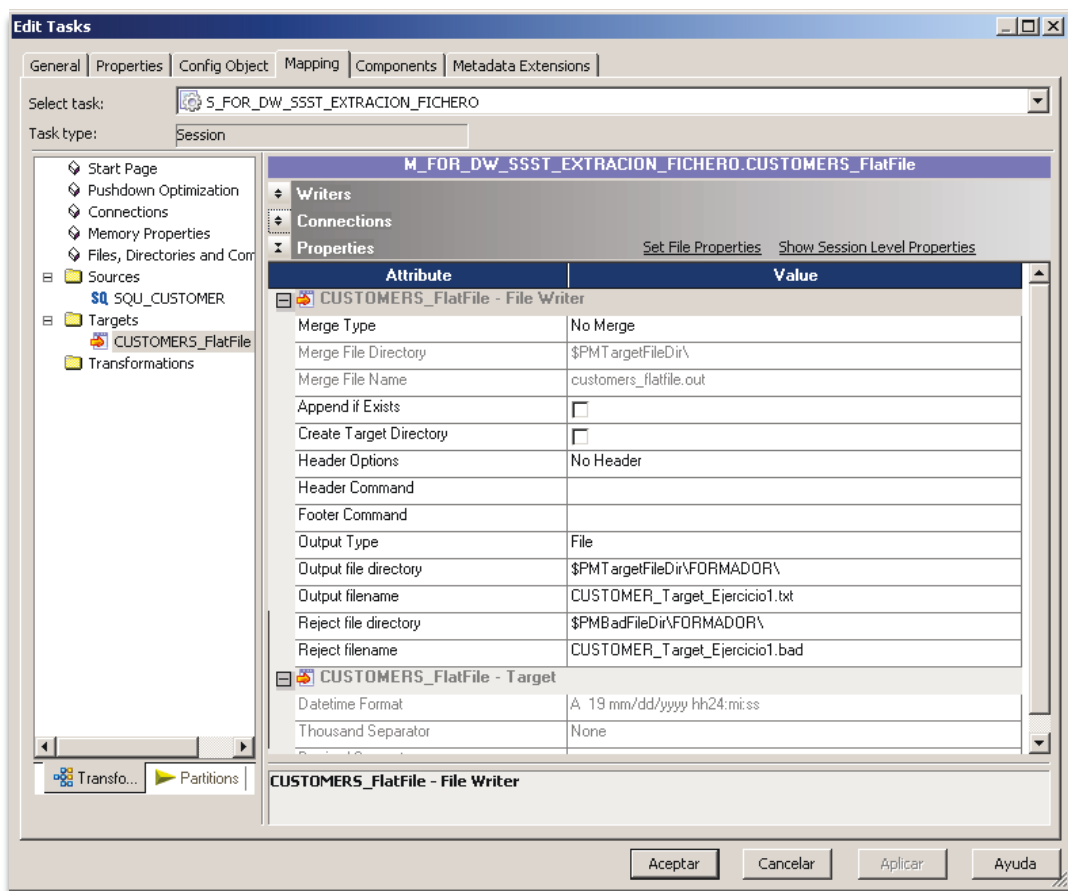


Figura 35: Pestaña Mapping – Targets – Fichero Plano

Opciones:

Opción	Descripción
Output file directory	Directorio de salida del fichero <code>\$PMTargetFileDir\[PROYECTO]\</code>
Output filename	Nombre del fichero (por ejemplo: CUSTOMER_Target.txt)
Reject file directory	Directorio de rechazados <code>\$PMBadFileDir\[PROYECTO]\</code>
Reject filename	Nombre del fichero de rechazados (con extensión .bad)

- Files, Directories and Commands

En este apartado, especificamos la carpeta donde se crearan los ficheros de memoria cache que utilizan algunas transformaciones, como por ejemplo, la transformación Joiner.

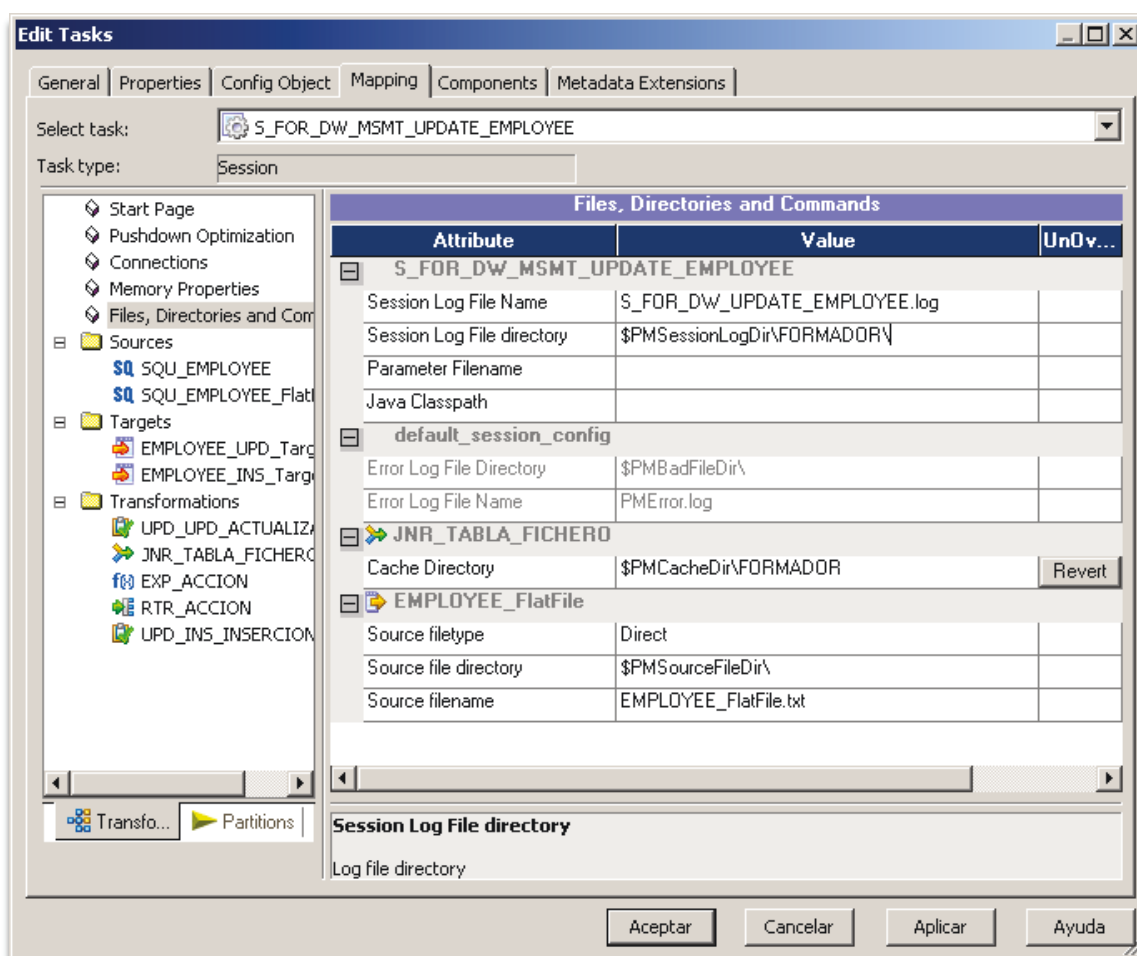


Figura 36: Files, Directories and Commands

Para las transformaciones que utilicen memoria cache:

Opción	Descripción
Cache Directory	Directorio de salida del fichero \$PMCacheDir\[PROYECTO] (no poder la última \)

e. Pestaña Components

La siguiente figura muestra la pestaña *Components* de una sesión que contiene tareas Comand.

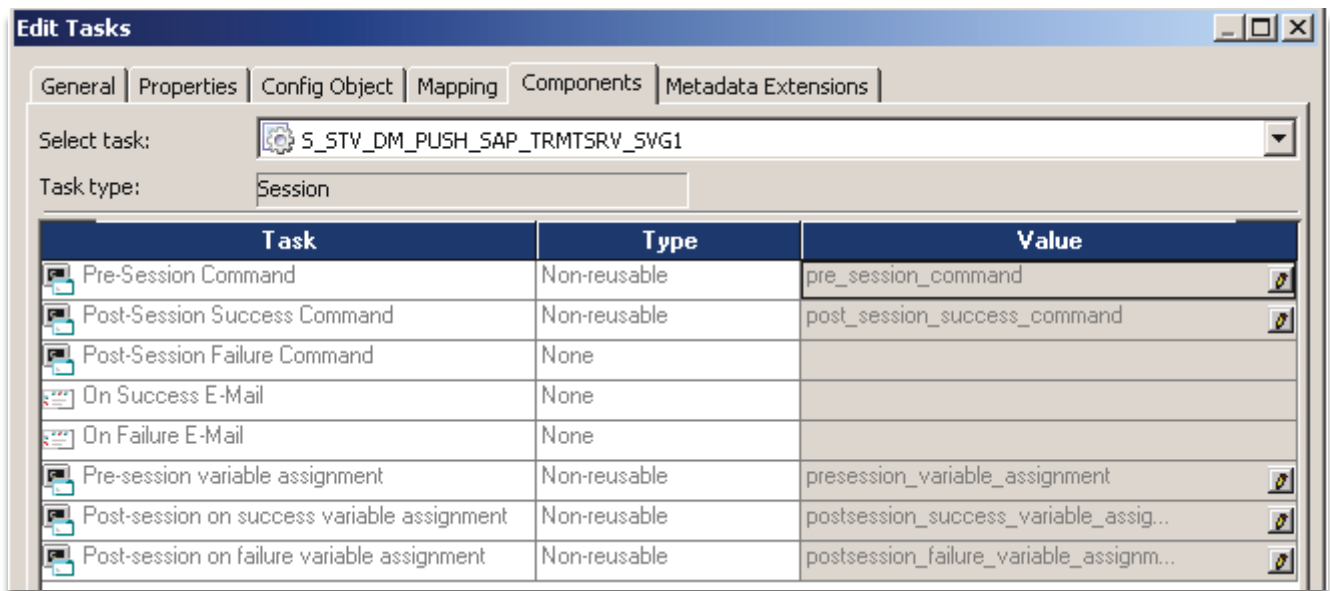


Figura 37: Pestaña Components

En esta pestaña se pueden ejecutar scripts a través de comandos (de borrado o truncado de tablas por ejemplo) antes del inicio de la sesión o al finalizar la sesión.

f. Validación de la sesión

Las sesiones se construyen sobre mappings validados previamente. Sin embargo si algún mapping sufre algún cambio hay que volver a validar la sesión correspondiente.

3.13. Ejecución de workflows y sesiones

➤ Ejecutar un workflow:

Para ejecutar un workflow, tiene que estar validado y sin errores en ningún componente. Hay que prestar especial atención al estado de las sesiones ya que el workflow puede ejecutarse teniendo una sesión impactada (no inválida) por algún cambio realizado y por lo tanto el resultado funcional puede que no sea el esperado.

Se puede ejecutar un workflow desde diferentes sitios.

- Haciendo clic sobre el icono en el figura siguiente. Esto ejecuta el workflow en su totalidad desde la tarea *Startk* hasta el final.
- Desde el Monitor como veremos más adelante.

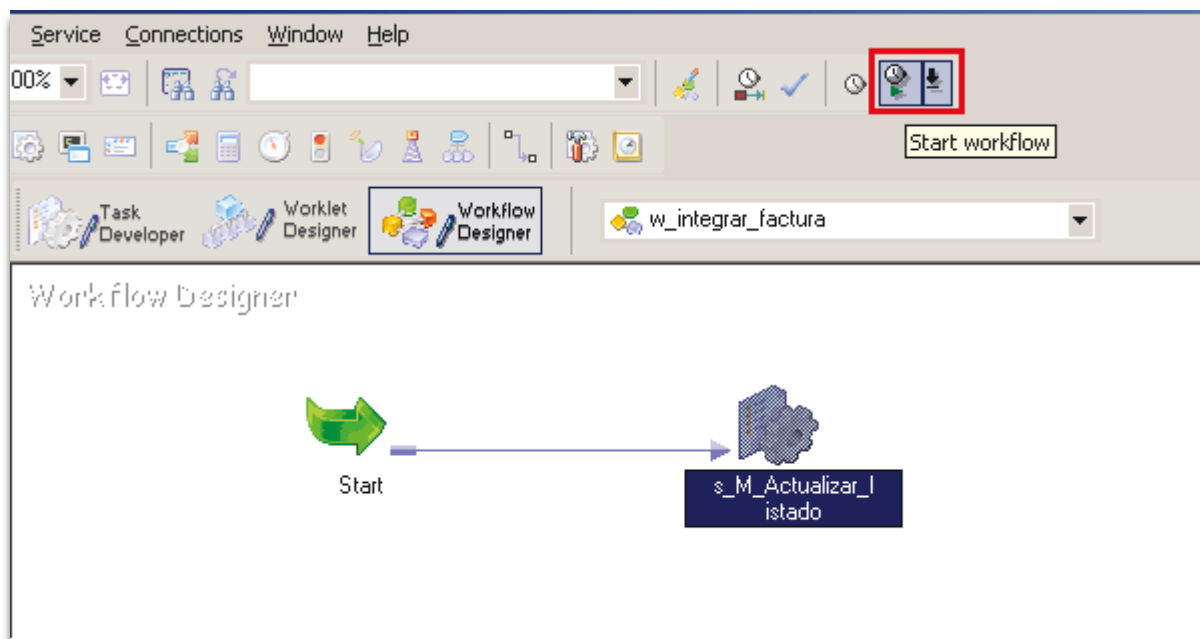


Figura 38: Ejecutar un workflow entero

➤ Ejecutar una sesión:

A veces nos interesa ejecutar una sola sesión dentro de un workflow. Por ejemplo, en un proyecto de mantenimiento de una aplicación que realiza operaciones ETL, si falla una sesión por falta de un fichero de entrada, para volver lanzar el workflow entero que puede tener muchas sesiones, consumiría mucho tiempo y podría tener bloqueada alguna parte de la aplicación. La solución sería lanzar manualmente sólo la sesión en cuestión o el resto de sesiones a partir de ésta.

Para lanzar una sesión: Botón derecho > *Start Task*.

Para lanzar el workflow a partir de una sesión: *Start Workflow From Task*

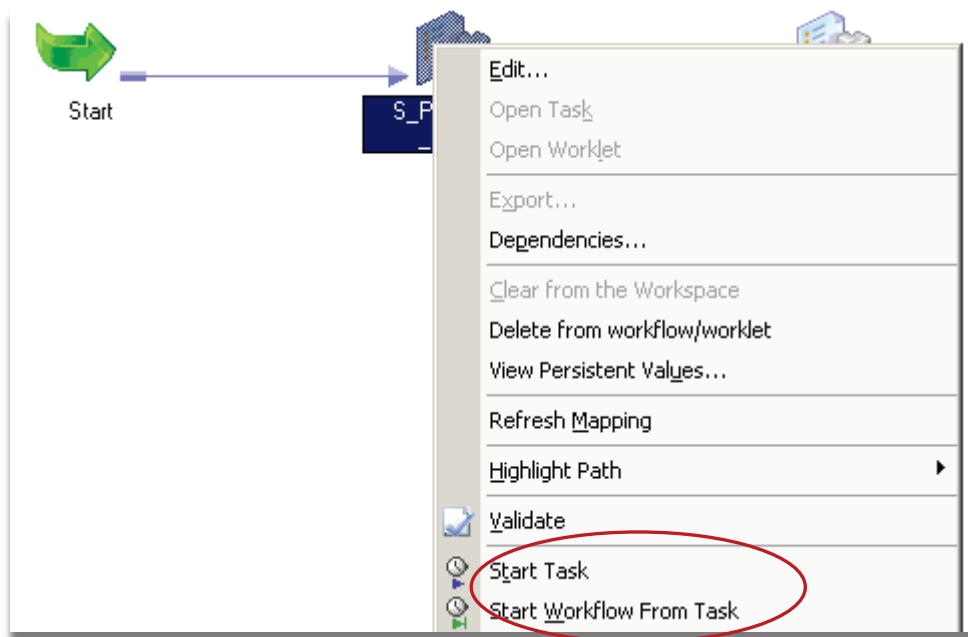


Figura 39: Ejecución de una sesión

3.14. Importación y Exportación de Objetos

Finalmente veremos cómo se exportan los objetos como workflows, worklets, Sources, Targets, etc. Se puede exportar un objeto en XML para utilizarlo para documentación, scripts de validación, etc.

Al exportar un workflow se exportan todos sus componentes. Al importarlo en un repositorio o carpeta donde exista algún objeto con el mismo nombre que un componente del workflow importado, hay que resolver algunos conflictos que aparecen en una pantalla. Se puede renombrar los objetos a importar o reutilizar los existentes, de esta manera no se pisan entre ellos para evitar un impacto sobre los workflows que utilizan los mismos objetos.

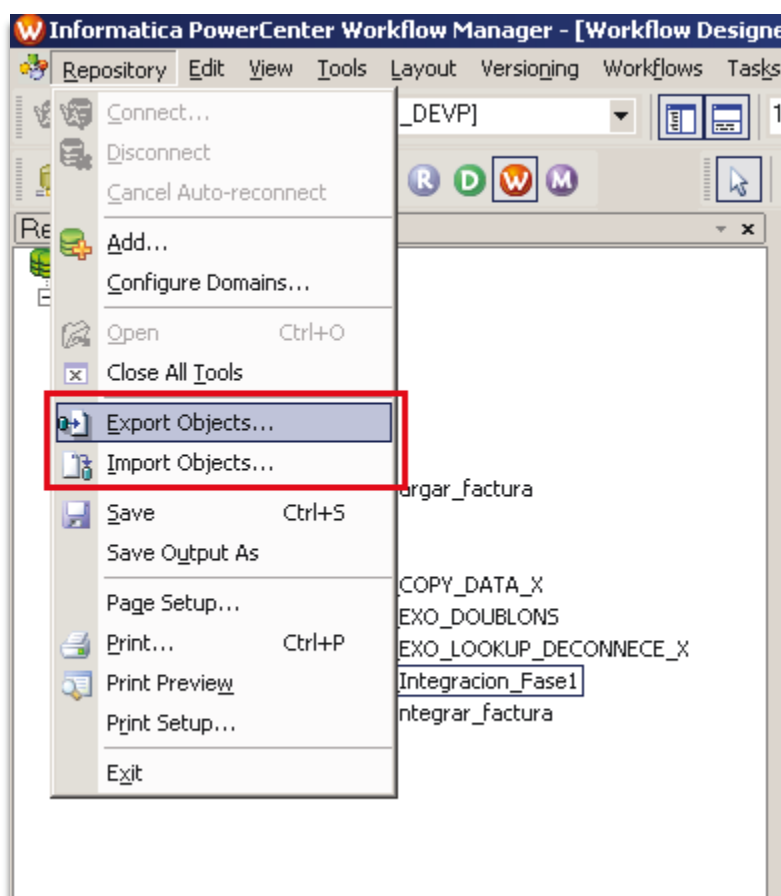


Figura 43: Importar/Exportar objetos

3.15. Workflow Monitor

3.15.1. Estado de ejecución

En el Monitor podemos visualizar las ejecuciones de los workflows de las diferentes carpetas y repositorios. Aquí podemos ver el transcurso de la ejecución así como información de los registros cargados en las tablas, tiempo de ejecución, estado, etc.

Vista de la herramienta Workflow Monitor:

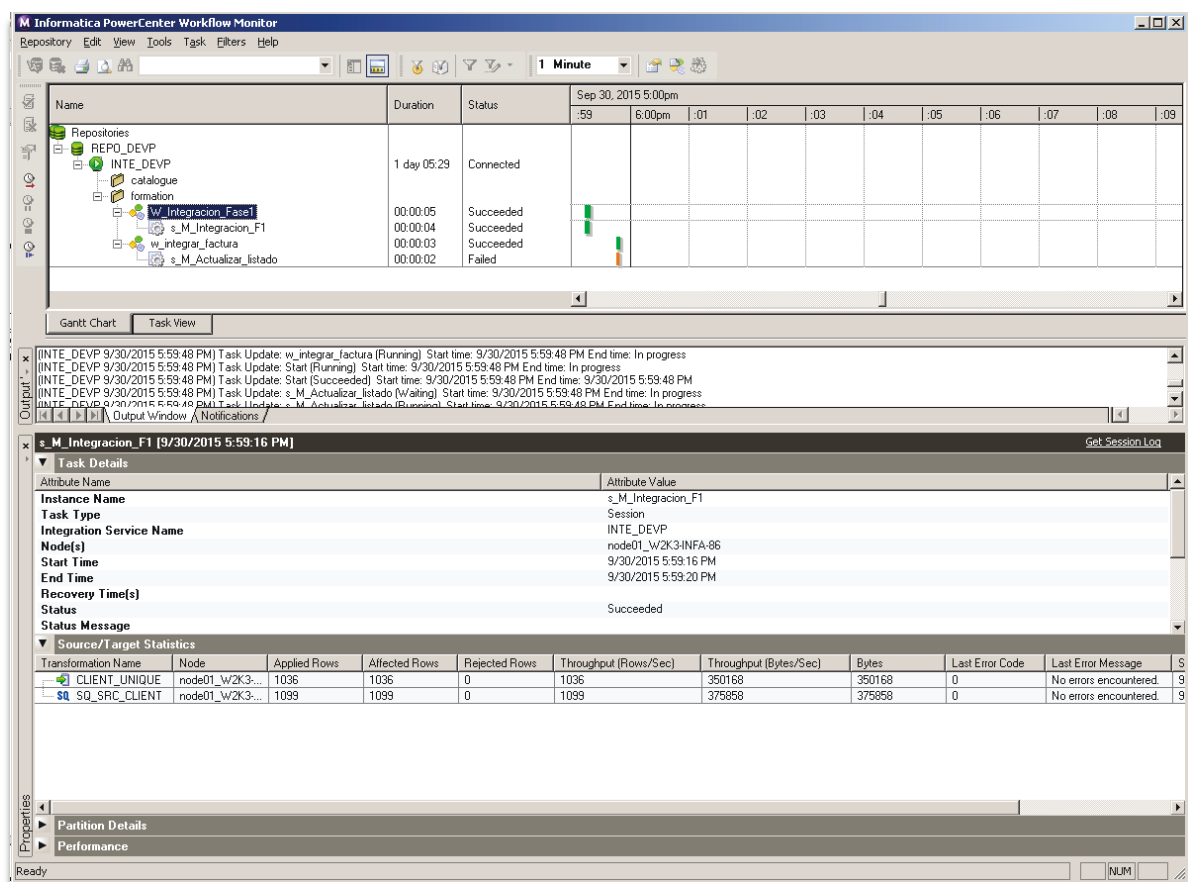


Figura 40: Workflow Monitor

Podemos observar en la columna Status, el estado de la ejecución. El primer workflow se ha ejecutado con éxito y el estado a "Succeeded", mientras que el segundo ha fallado y el estado está a "Failed".

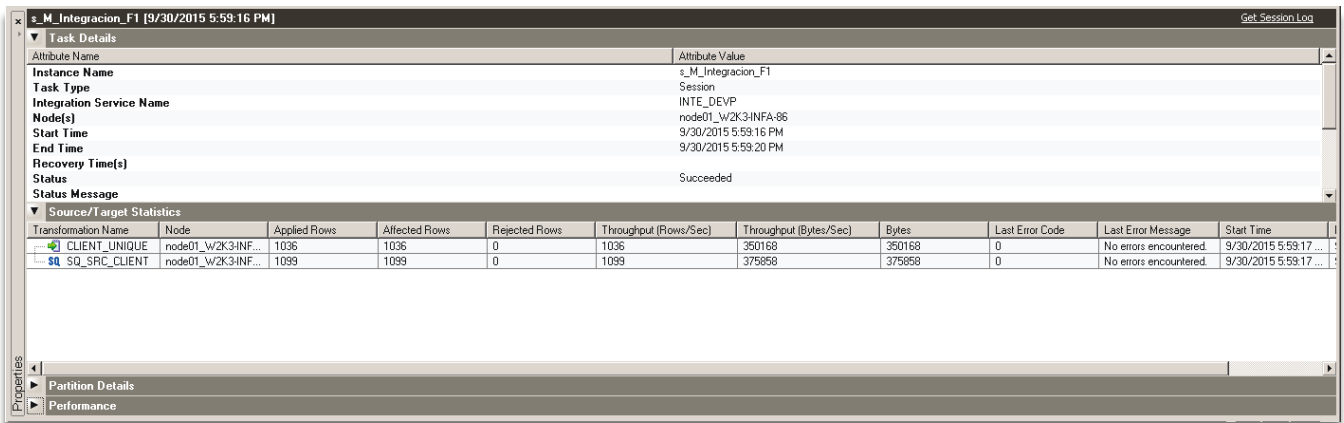
Además de consultar el estado de la ejecución, se puede relanzar los workflows o las sesiones a partir del Workflow Monitor.

3.15.2. Propiedades de la ejecución

El Workflow Manager proporciona información sobre la ejecución como fecha de ejecución, hora, duración, SI, etc. Lo más importante es la información relativa a los datos tratados tanto en el Source como en el Target.

En la ventana “*Source/Target Statistics*” podemos observar el número de registros extraídos de la fuente y los insertados en el destino, registros rechazados, número de registros tratados por segundo, códigos de errores, etc.

Esto es que lo ayuda al analista de comprobar la coherencia del proceso ejecutado, antes de realizar más pruebas a nivel de BDD.



The screenshot shows the 'Task Details' window for a session named 's_M_Integracion_F1' [9/30/2015 5:59:16 PM]. The 'Task Details' section shows the following information:

Attribute Name	Attribute Value
Instance Name	s_M_Integracion_F1
Task Type	Session
Integration Service Name	INTE_DEVP
Node(s)	node01_W2K3-INF-06
Start Time	9/30/2015 5:59:16 PM
End Time	9/30/2015 5:59:20 PM
Recovery Time(s)	
Status	Succeeded
Status Message	

The 'Source/Target Statistics' section shows the following data:

Transformation Name	Node	Applied Rows	Affected Rows	Rejected Rows	Throughput (Rows/Sec)	Throughput (Bytes/Sec)	Bytes	Last Error Code	Last Error Message	Start Time
CLIENT_UNIQUE	node01_W2K3-INF...	1036	1036	0	1036	350168	350168	0	No errors encountered.	9/30/2015 5:59:17 ...
SQ_SRC_CLIENT	node01_W2K3-INF...	1099	1099	0	1099	375858	375858	0	No errors encountered.	9/30/2015 5:59:17 ...

Figura 41: Propiedades de la ejecución de la sesión

3.15.3. Log de la sesión

El Log de la ejecución es crucial a la hora del análisis de fallos ya que ahí encontramos todo tipo de información sobre el proceso, incluidos los errores producidos durante la ejecución.

En general, el log de una sesión contiene:

- Información sobre el directorio donde se crea el log, repositorio, carpeta, servidor, nombre de Workflow y Mapping asociado.

- Valor que toman los parámetros definidos en el Mapping y en el Workflow.
- Ejecución de comandos pre-sesión.
- Consultas SQL de las transformaciones que aparecen en el Mapping.
- Tiempos de inicio y fin del proceso.
- Errores y avisos encontrados durante la sesión.
- Ejecución de comandos post-sesión.
- Resumen de lecturas en Source Qualifier y Targets y Estadísticas DTM.

Clicando con el botón derecho del ratón sobre la sesión, podemos ver el log de la sesión, tanto si ha fallado como si se ha ejecutado correctamente. En este caso vamos a consultar el log de ejecución de una sesión fallida.

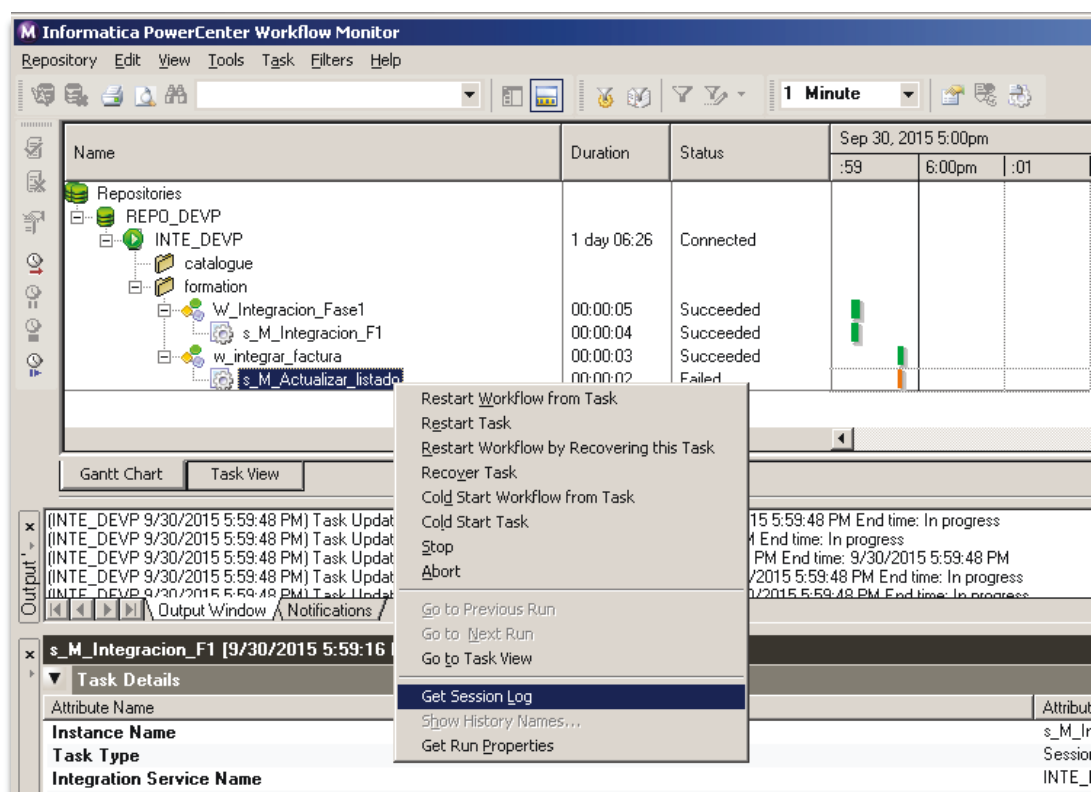


Figura 42: Abrir el log de la sesión

La siguiente vista muestra el log de la sesión:

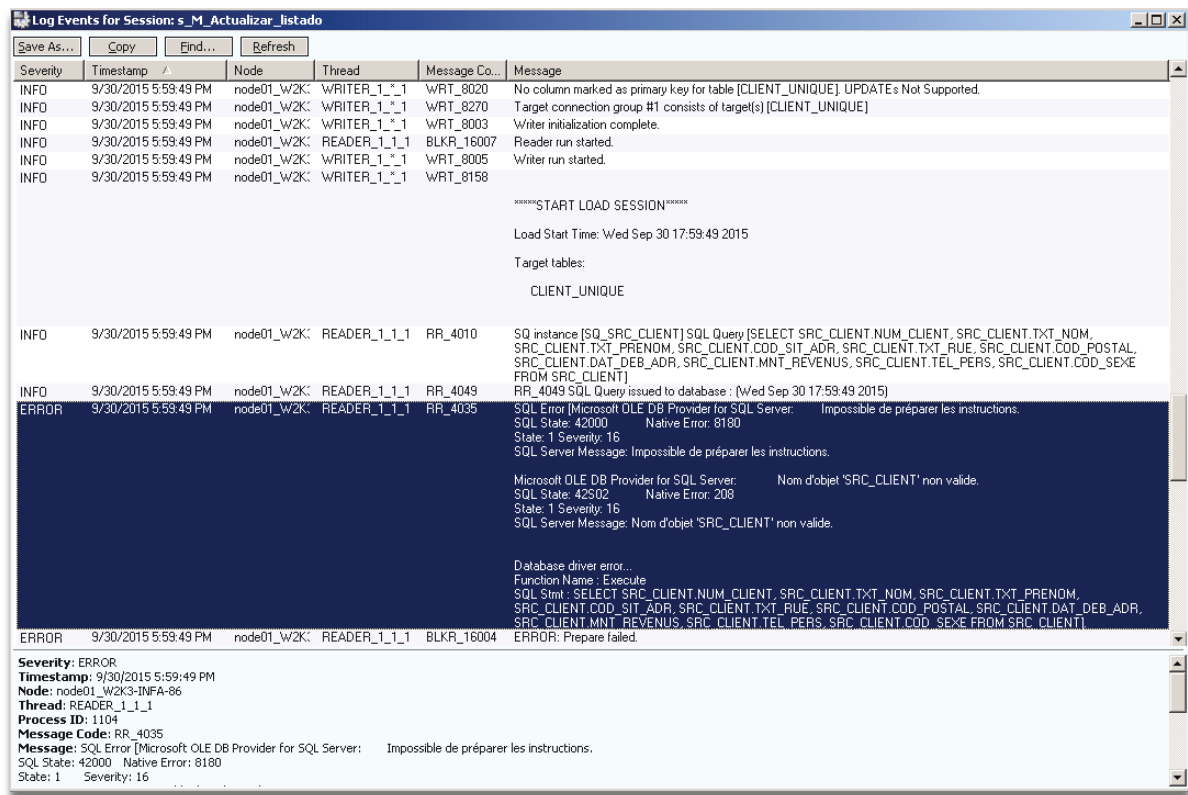


Figura 43: Log de la sesión

4. Implementación de casos prácticos

4.1. Introducción

Después de haber visto una descripción del funcionamiento del Workflow Manager de Powercenter y otras buenas prácticas, ahora vamos a implementar unos casos prácticos para ver de cerca algunas soluciones que responden a algunos requisitos del cliente.

En este tipo de proyectos entran muchas variables como, plataformas, herramientas, bases de datos, etc. en función de la envergadura y naturalidad de los mismos. Aquí nos centraremos en lo que se refiere a las soluciones que ofrece Powercenter como diseño de workflows, etc.

Las tecnologías que utilizaremos son las detalladas en el punto 2 (Estado del arte).

4.2. Requisitos

Un cliente importante multinacional nos pide diseñar nuevos procesos de ETL para una aplicación ya en funcionamiento para controlar sus ventas. Nos consta que disponemos de las siguientes herramientas:

- Informatica Powercenter 9.1 instalado en un servidor Linux.
- Base de datos Oracle 11g que constituye el ODS y el DWH.
- PowerExchange con varios conectores de tipos de datos.

Los tratamientos tienen como misión la extracción de datos de ficheros provenientes de una base de datos transaccional, realizar una serie de transformaciones definidas por los analistas funcionales del cliente, guardarlos en el ODS y exportarlos al DWH.

El cliente nos especifica los siguientes requisitos:

- Guardar un histórico de los datos extraídos.
 - Información histórica de ventas por regiones.
- La extracción tiene que realizarse en el menor tiempo posible para dejar el DWH disponible para otras consultas.
- Una medida para que el impacto del aumento del histórico de datos sea lo menor posible para no ralentizar la aplicación.
- Los procesos tienen que estar diseñados de manera a no tener que configurarlos en cada extracción, ya que los ficheros de entrada Mainframe tendrán nomenclaturas diferentes con fecha del día.

4.3. Soluciones propuestas

Del análisis realizado y las tecnologías de las que disponemos, proponemos las siguientes soluciones a los requisitos especificados:

- El histórico de datos se guardará en la base de datos del ODS en una tabla particionada. Esta tabla estará particionada por regiones (código de región), ya que la versión de Oracle que tenemos lo permite.
El tipo de particionamiento será: *PARTITION BY LIST* por el código de región.
- Los workflows se diseñarán de manera a aprovechar en la medida de lo posible el particionamiento de la tabla para que el proceso de carga dure menos tiempo.

Primero extraemos los datos brutos de los ficheros fuente y se guardarán en una tabla intermedia sin apenas transformaciones de datos. Esta tabla se vaciará antes de cada carga.

La segunda fase consiste en cargar los datos en la tabla del histórico. Por último, en la tercera fase se guardarán únicamente los datos del mes corriente en una tabla final que será accesible por el negocio.

4.4. Implementación

4.4.1. Fase 1.

En esta fase desarrollaremos los workflows de extracción de datos de los ficheros de entrada.

Los ficheros de entrada están en formato Cobol y otros en formato FlatFile. Para los primeros hay que utilizar la herramienta PowerExchange para construir las estructuras Datamap de las fuentes que utilizará Powercenter.

Estructura del fichero de entrada Cobol:

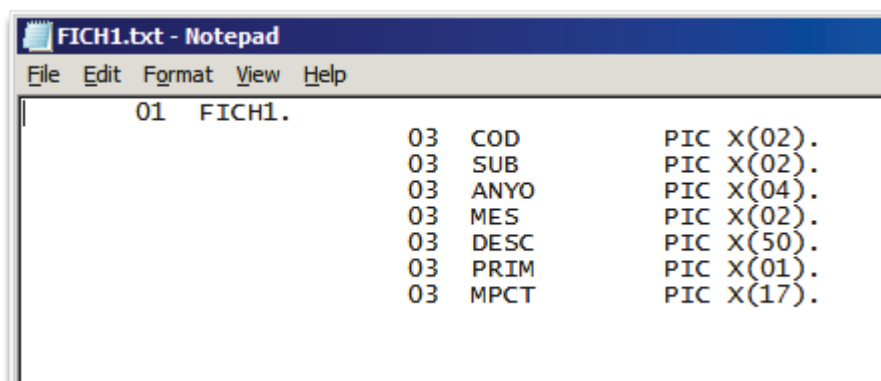


Figura 44: Estructura de fichero Cobol

En PowerExchange convertiremos la estructura del fichero Cobol en un Datamap que construirá la estructura fuente.

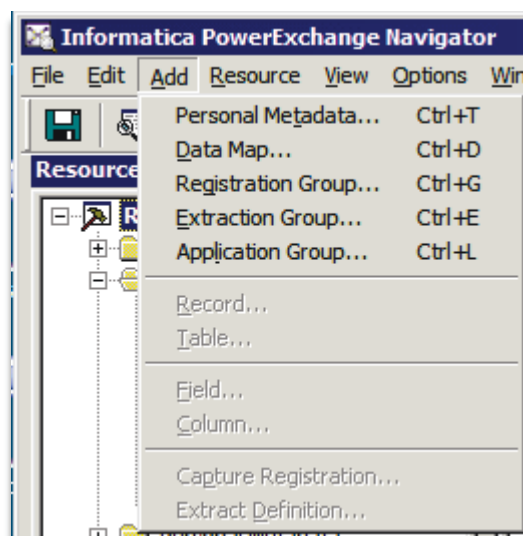


Figura 45.1: Crear Datamap

Haciendo clic en Add > Data Map ...

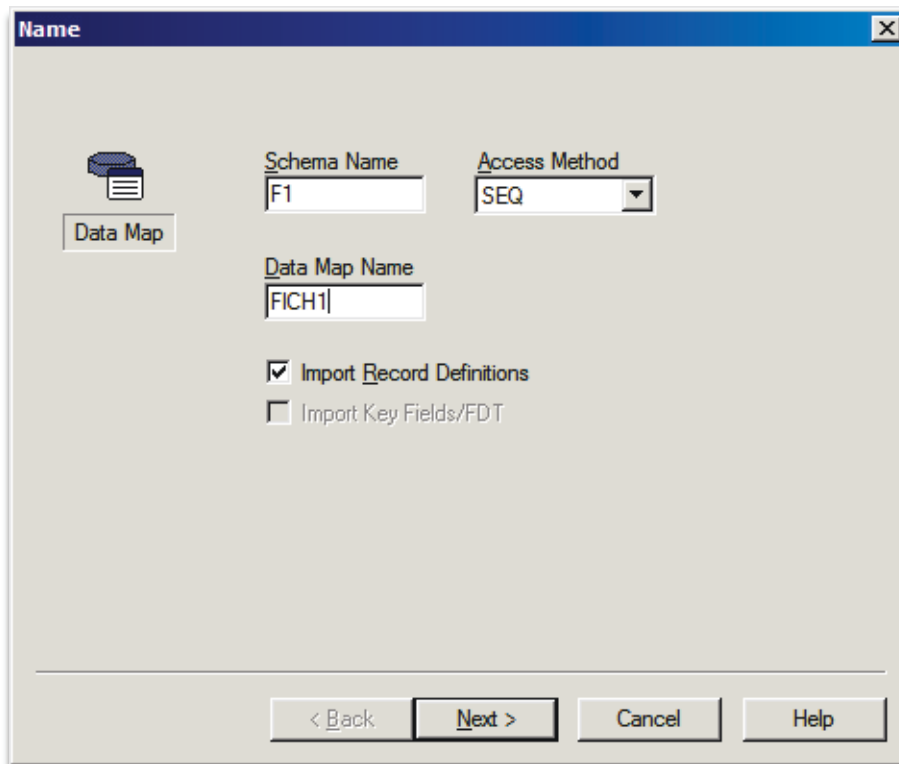


Figura 45.2: Crear Datamap – Nombre fichero

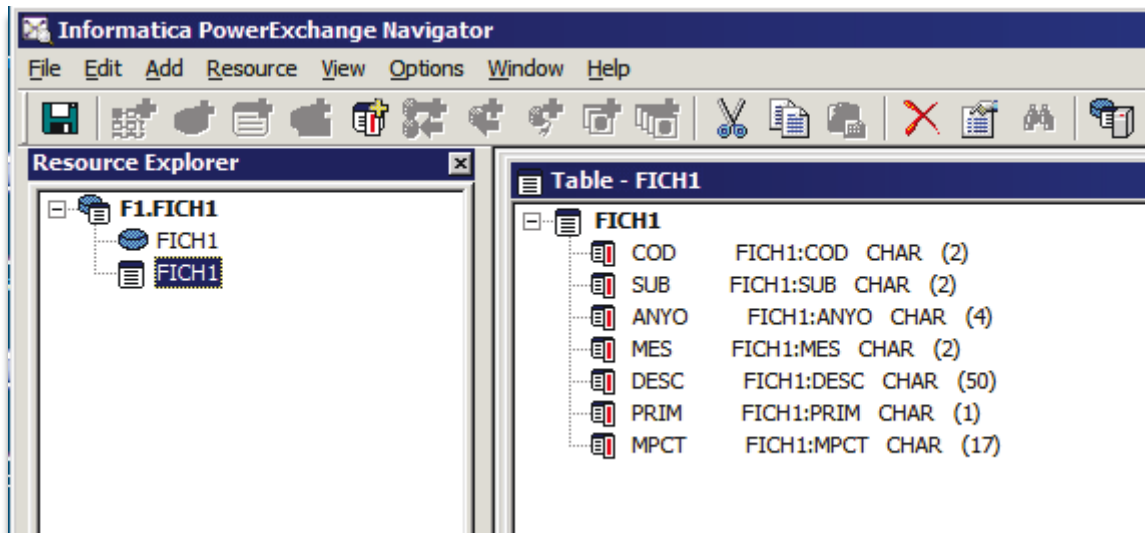


Figura 45.3: Estructura del Datamap creado

Podemos observar cómo PowerExchanger ha convertido la estructura del fichero Cobol a un formato de tabla con campos de tipo CHAR() en vez de PIC

X()). Esta Estructura se importará desde Powercenter para ser utilizada como Source.

a. Workflow w_Extract_Cob_Files

Ahora procedemos a crear el workflow sobre el mapping que contiene algunas transformaciones de los tipos de datos.

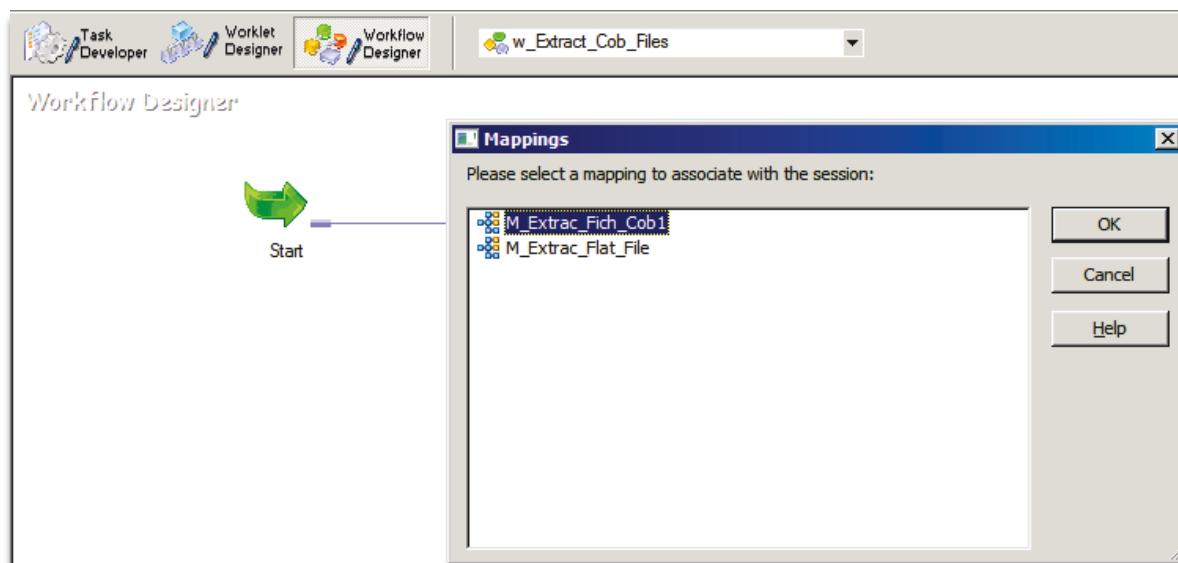


Figura 46: Crear workflow de extracción de fichero Cobol

Configuración de la sesión:

Seleccionamos la opción "Fail parent if this task fails" para parar el workflow en caso de error a nivel de esta sesión.

Seleccionamos las conexiones del fichero fuente y el esquema de la tabla destino.

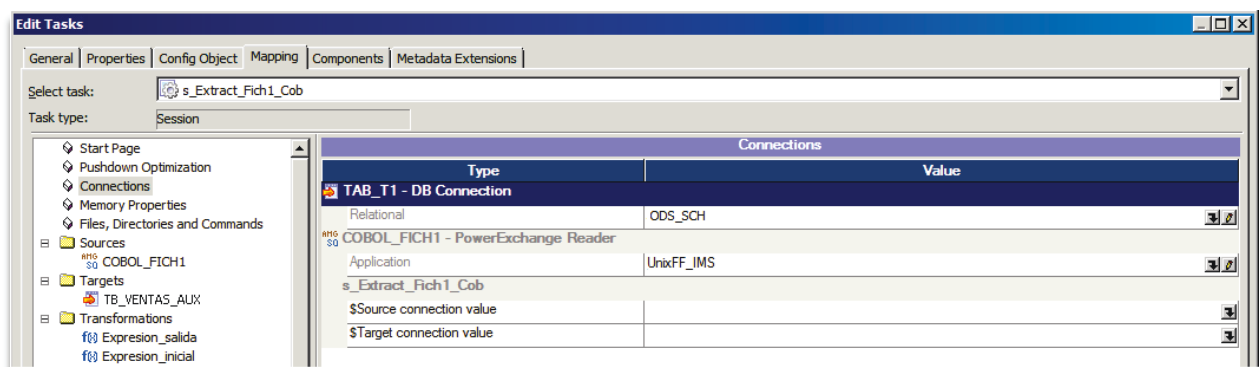


Figura 47: Configuración de conexiones workflow 1

Configuramos el fichero de entrada. Dado que los ficheros Mainframe tendrán nombres diferentes cada día, no podemos establecer un nombre fijo. Para ello utilizaremos una lista que contendrá las rutas seguidas de los ficheros fuente. Powercenter accederá siempre a lista primero, que tendrá un nombre fijo (COB_FILES.lst).

Seleccionamos la opción "Filelist File" y añadimos el directorio parametrizado donde se encontrará la lista de ficheros.

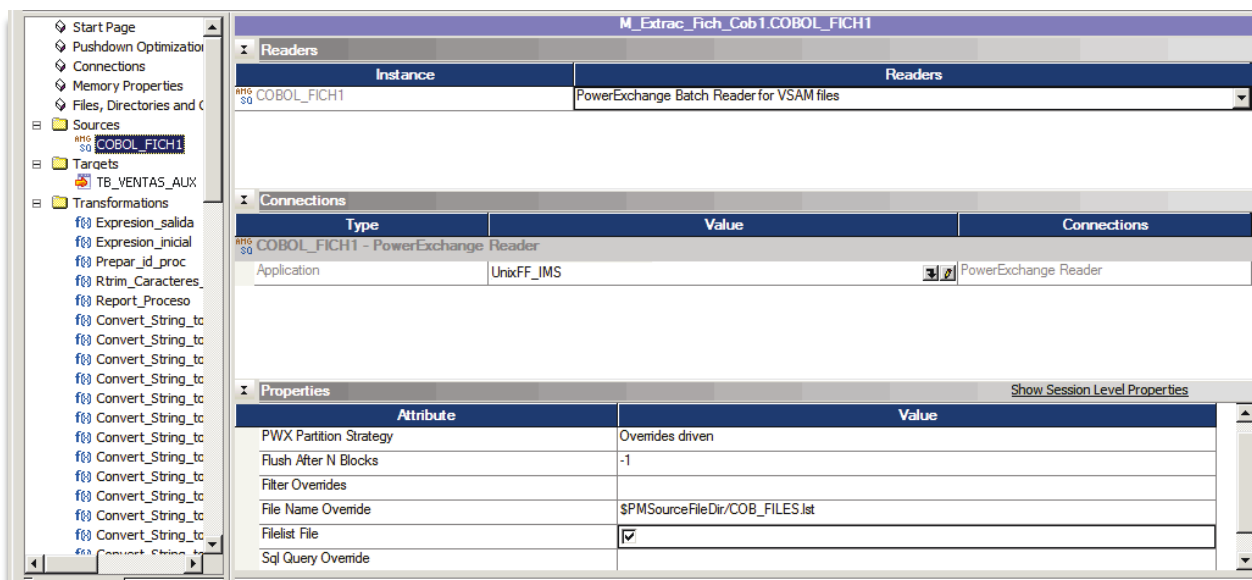


Figura 47: Configuración del fichero de entrada

En esta fase no vamos a guardar el histórico de datos, la tabla destino se truncará en cada extracción. Para ello seleccionamos la opción "Truncate

target table option". Dado que no hay operaciones de UPDATE, el tipo de carga será "Bulk" con el fin de realizar la carga de datos de manera algo más rápida.

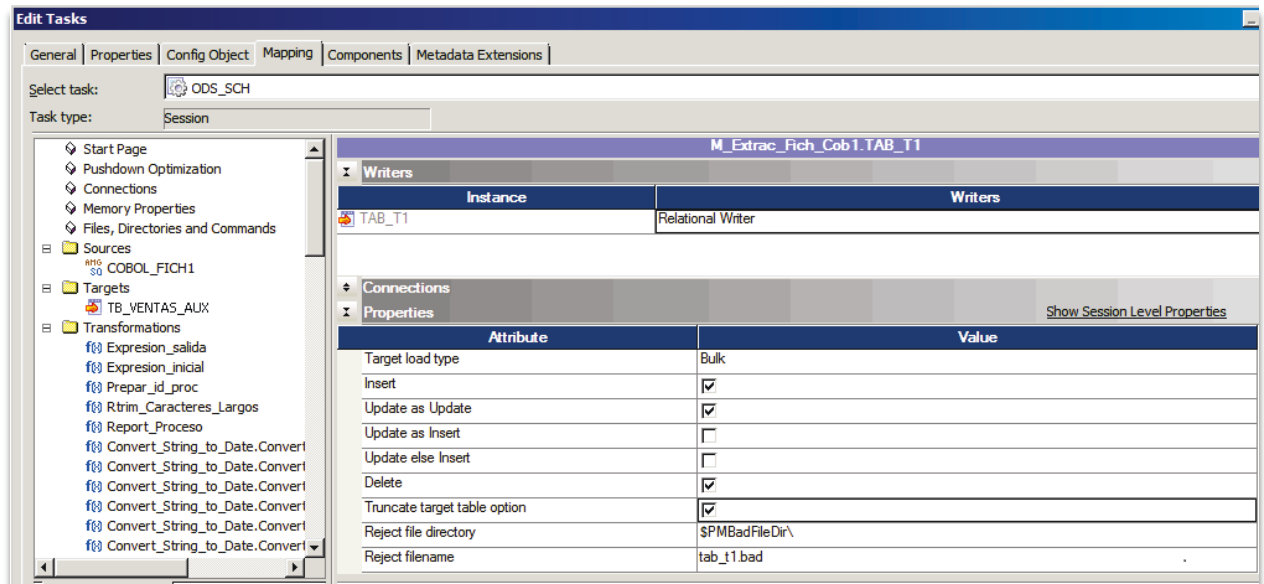


Figura 48: Configuración de la tabla destino.

Finalmente el workflow es de una sola sesión para la extracción del primer fichero Cobol.

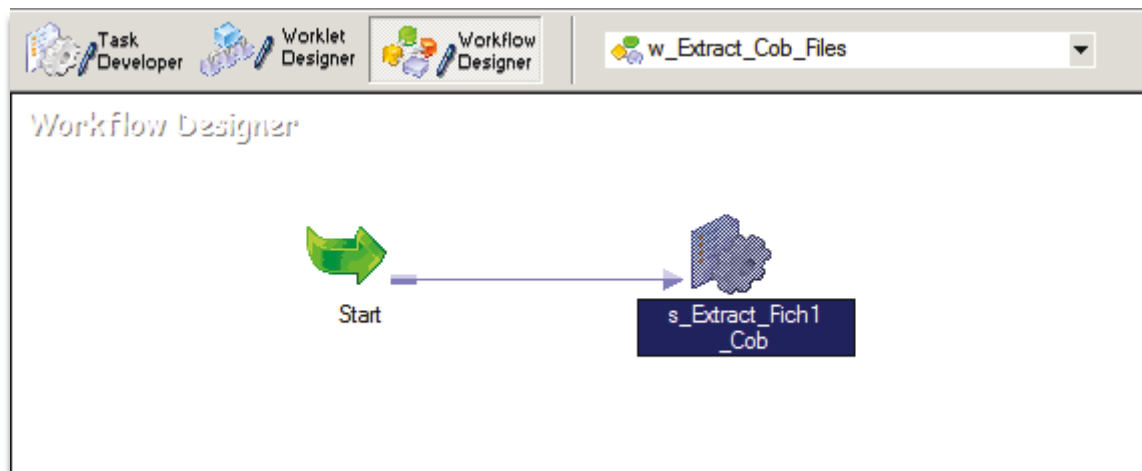


Figura 49: Workflow Extract Cob Files

b. Workflow w_Extract_Flat_Files

Este workflow extrae datos de los ficheros planos que tendremos de entrada. Los ficheros tendrán siempre el mismo nombre en el servidor, por lo que creamos una sesion por fichero, configuradas con las conexiones y nombres de ficheros fijos.

En configuración de los ficheros de entrada, seleccionamos la opción “*Source filetype = Direct*” ya que los ficheros tendrán la misma nomenclatura en cada extracción y no se utilizarán listas de ficheros.

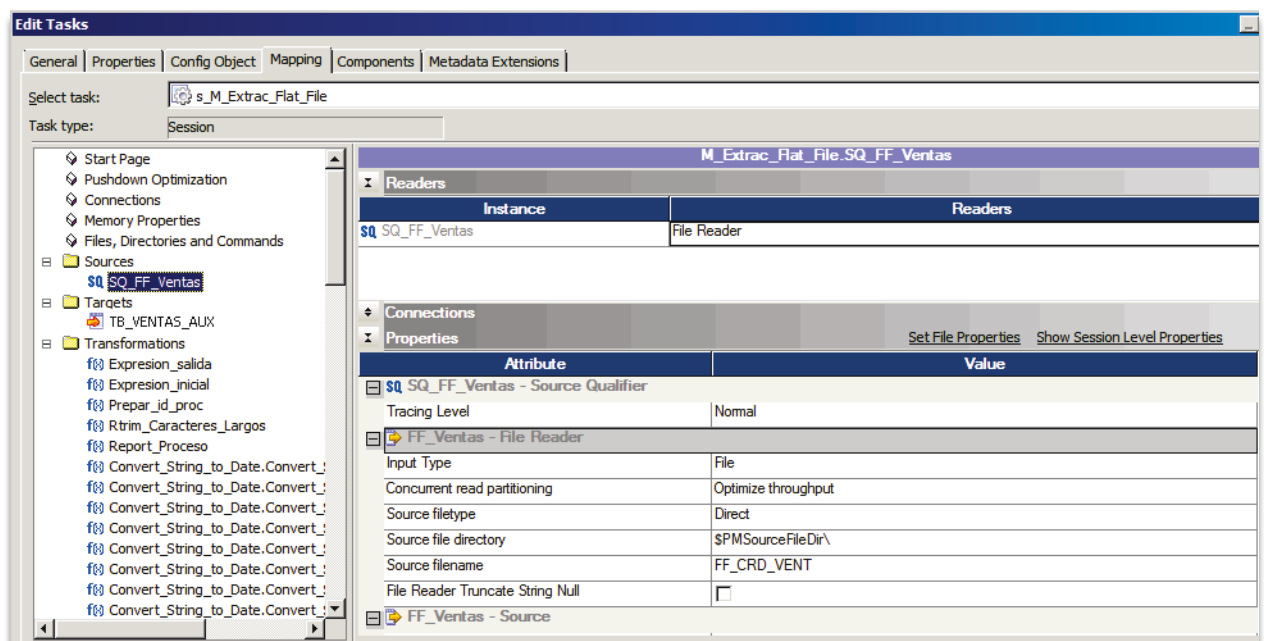


Figura 50: Configuración del fichero plano.

Esta vez la tabla no se truncará para no borrar la información cargada por el primer workflow, por lo que no seleccionamos la opción “*Truncate target table option*” y el tipo de carga “Bulk” por no tener operaciones de Update.

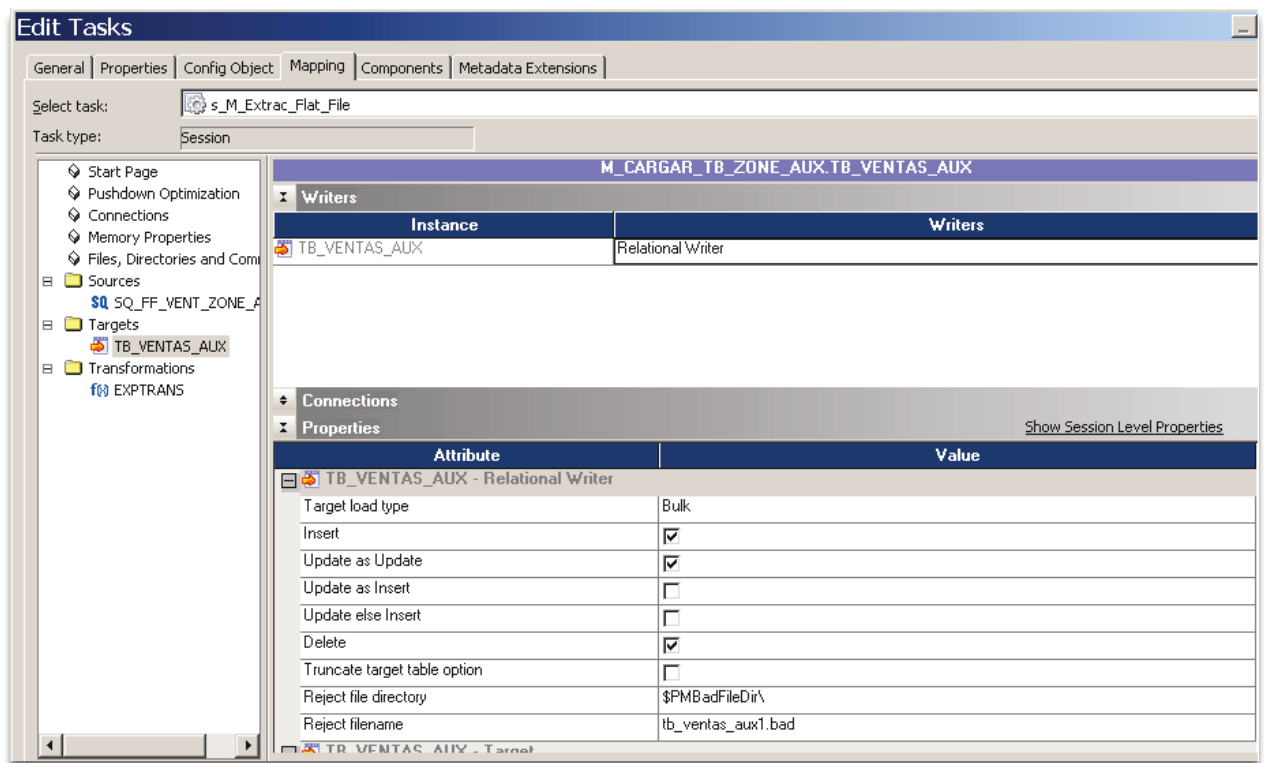


Figura 51: Configuración de la tabla destino.

La carga de datos de los ficheros se hace sobre la misma tabla destino "TB_VENTAS_AUX", por lo cual las sesiones se ejecutarán secuencialmente, extrayendo los datos de cada fichero y cargando en la tabla.

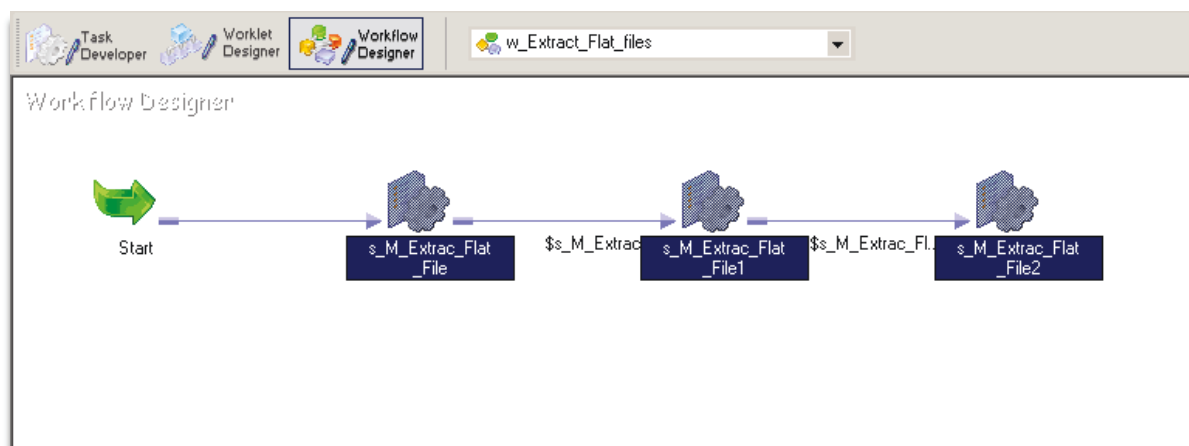


Figura 52: Workflow Extract Flat Files

Podemos observar como despues de haber validado y guardado los workflows, aparecen con el Check en verde indicando la validez de los mismos.

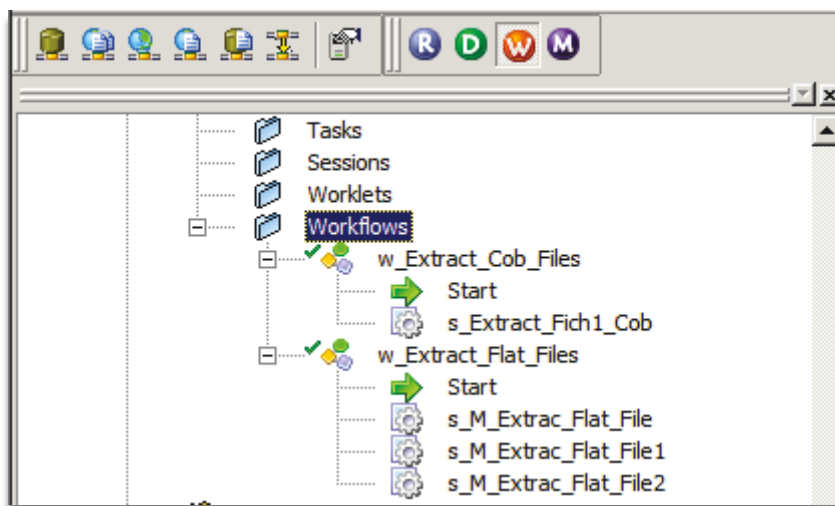


Figura 53: Workflows validados

Con esto acabamos la primera fase de extracción.

4.4.2. Fase 2.

En la primera fase hemos extraído los datos brutos sin apenas realizar cambios sobre los mismos. En esta fase se guardan, con algunos cambios, en las tablas particionadas que almacenan el histórico de datos.

El workflow de esta fase contendrá un mecanismo de control para respetar el orden de ejecución del resto de la cadena de workflows. Añadimos una tarea *Comand* que se encargará de crear un fichero de texto que servirá de flag para "avisar" al siguiente workflow de que ya puede ejecutarse.

El workflow de la siguiente fase tendrá que esperar a que se termine esta fase con éxito. De lo contrario, no se lanzarán.

También incluimos tareas de tipo *Decision* para controlar la ejecución de todas las sesiones que contiene el workflow, de manera que éste último termine su ejecución si y solo si todas las sesiones acaban con éxito. En caso de que alguna sesión falle, el workflow se detendrá con la tarea *Control*.

La siguiente imagen muestra el contenido del workflow "w_CARGAR_TB_ODS_TH".

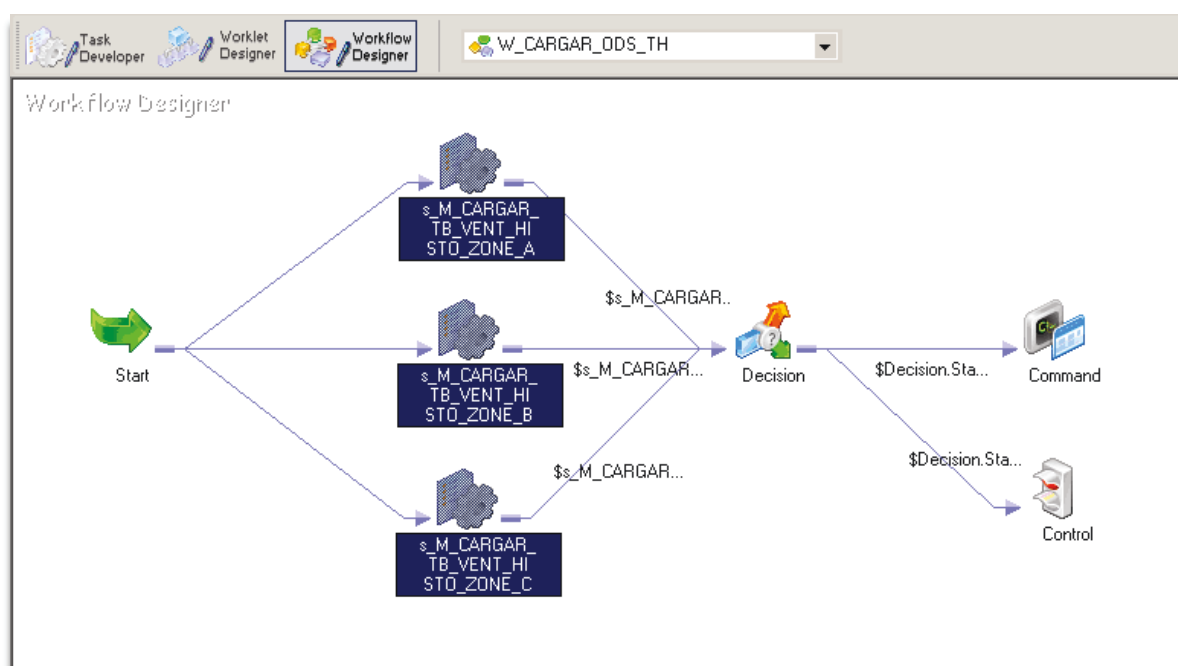


Figura 54: Workflow fase 2

Para optimizar el tiempo de carga de datos en la tabla de histórico, dado que su tamaño crecerá en cada carga, haremos una lectura en paralelo sobre la tabla fuente para escribir en paralelo en las particiones de la tabla del histórico.

Powercenter permite utilizar la misma sesión varias veces en el mismo workflow con diferente configuración. Esto permitirá incorporar una consulta SQL dentro de cada sesión para extraer diferentes rangos de datos en paralelo.

La tabla "TB_VENT_ZONE_TH" está particionada por código de zona (ZONE_ID = A, B o C). La configuración del SQ será como se ve a continuación:

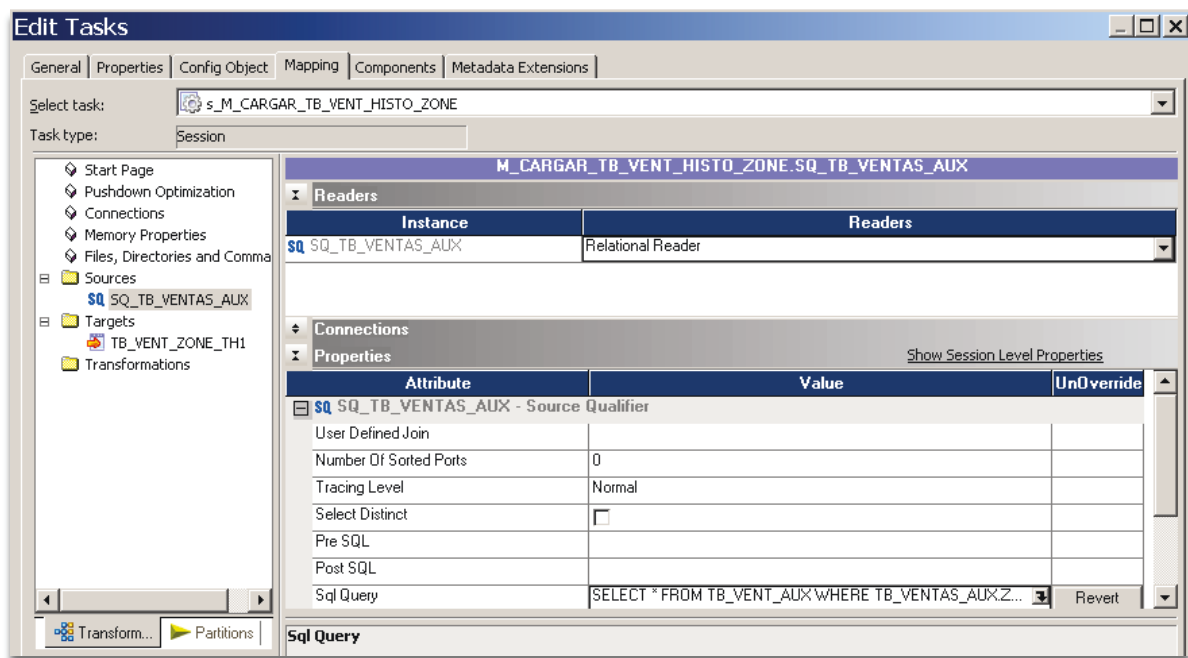


Figura 55: Sesión 1 Tabla "TB_VENTAS_AUX"

En el campo "Sql Query" añadimos la consulta que filtra los datos correspondientes a la zona "A" de la tabla "TB_VENTAS_AUX" de entrada.

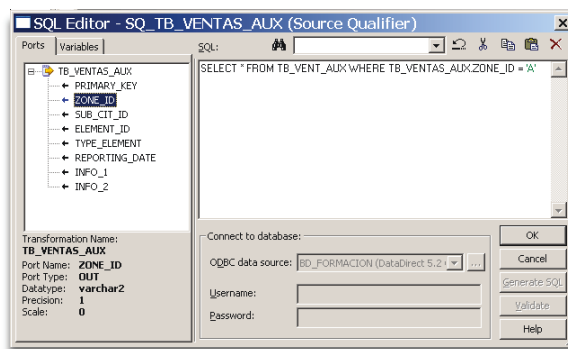


Figura 56: Query de la sesión Zone A

La segunda y tercera sesión tendrán la misma configuración que la primera, con la diferencia de la query incorporada en cada una de ellas como indican las siguientes imágenes.

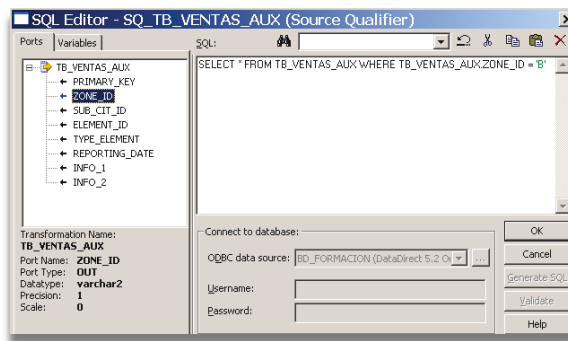


Figura 57: Query de la sesión Zone B

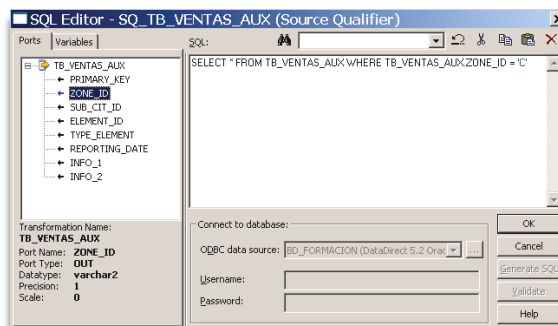


Figura 58: Query de la sesión Zone C

La siguiente imagen muestra la configuración que corresponde a la carga de datos en la tabla particionada "TB_VENT_ZONE_TH" en cada una de la sesiones. Esta tabla contiene el histórico de datos por lo que no se truncará.

Tampoco se realizarán operaciones Update, por lo que el tipo de carga de datos puede ser "Bulk". Otras configuraciones como, conexiones, fichero de registros rechazados también se hacen a este nivel.

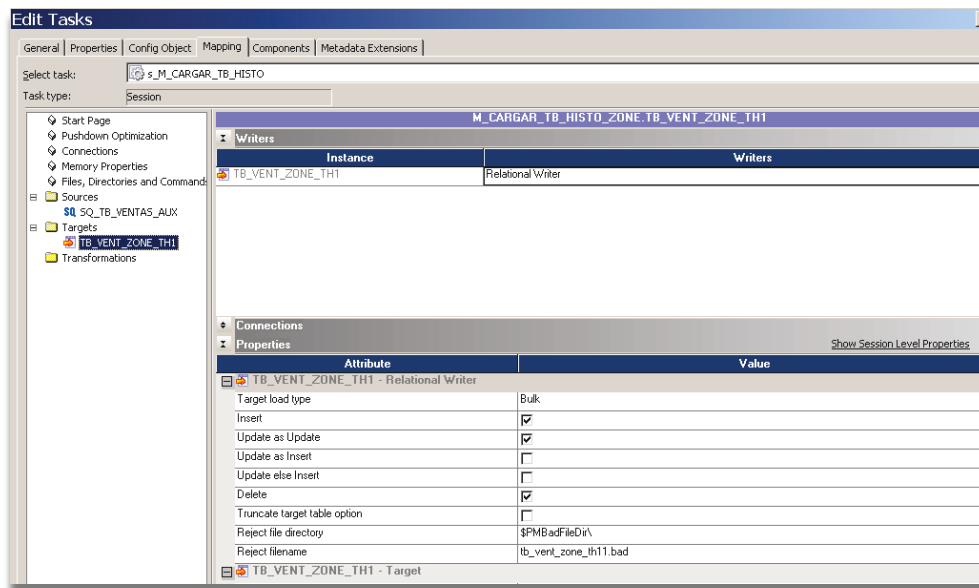


Figura 59: Sesión Tabla "TB_VENT_ZONE_TH"

Siempre seleccionando la opción "Fail parent if this task fails" para detener el workflow en caso de fallo de sesión.

Ahora procedemos a definir el flujo de la ejecución dentro del workflow. Los enlaces que unen la tarea "Decision" contendrán condiciones que decidirán si ejecutar la siguiente tarea o abortar la ejecución del workflow.

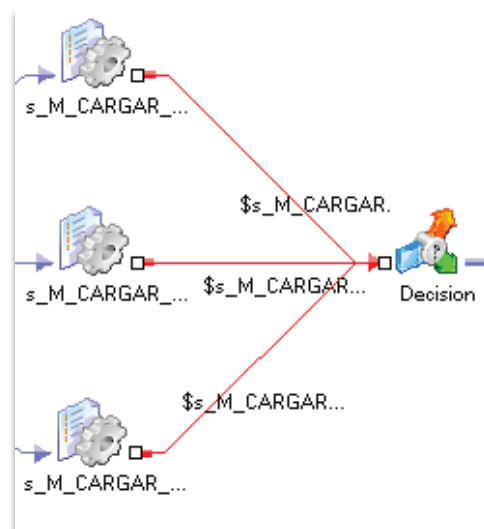


Figura 60: Entrada de la Decision

El primer enlace que une la *Decision* y la siguiente sesión, tendrá la condición de que la sesión anterior se haya ejecutado con éxito. En las siguientes imágenes podemos observar la condición del estado de cada una de las sesiones anteriores (utilizando los parámetros de Powercenter) que tienen que tener el valor "Succeeded" indicando que la ejecución ha terminado con éxito.



Figura 61.1: Condición del primer enlace

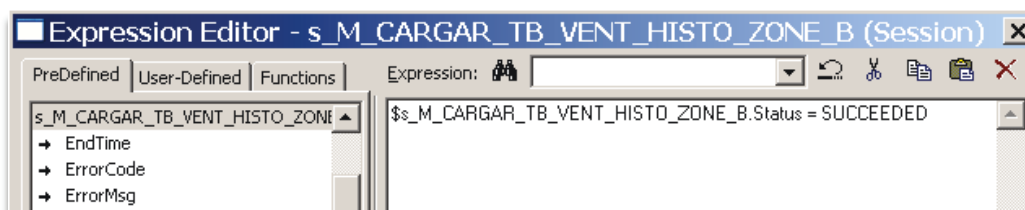


Figura 61.2: Condición del segundo enlace



Figura 61.3: Condición del segundo enlace

En la *Decision* añadimos la condición que exige la ejecución exitosa de las sesiones anteriores para seguir el flujo de ejecución.

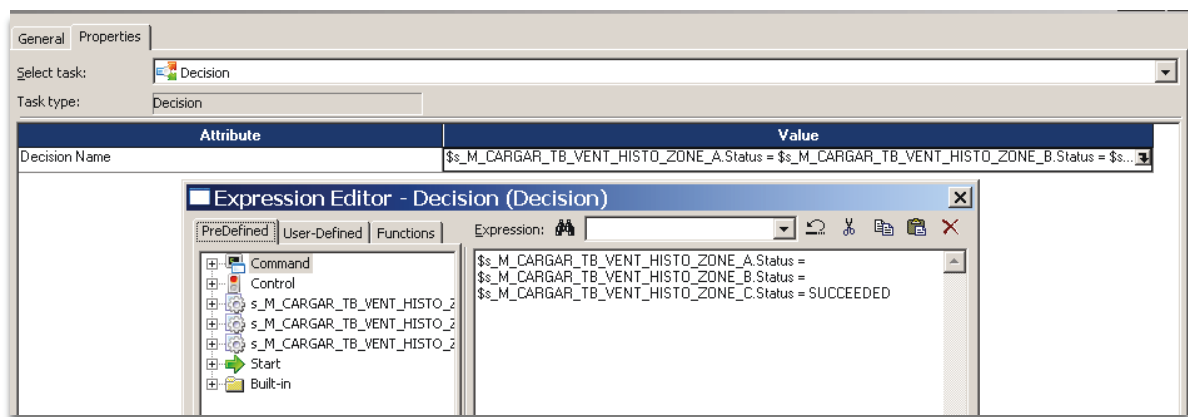


Figura 62: Condición de la Decisión

A la salida de la Decisión el primer enlace conecta con la última tarea del workflow (*Command*) con la condición “*\$Decision.Status = SUCCEEDED*”.

El segundo enlace conecta con la tarea *Control* para abortar el worflow en caso de error. Contiene la condición “*\$Decision.Status != SUCCEEDED*” indicando que la sesión *Decision* ha terminado con error.

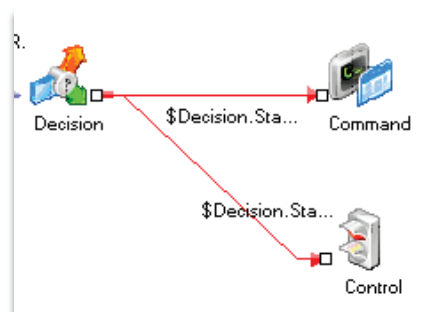


Figura 63: Salida de la Decisión

La tarea *Control* tendrá como objetivo abortar la ejecución del workflow, por lo que tendrá la opción “Fail parent”.

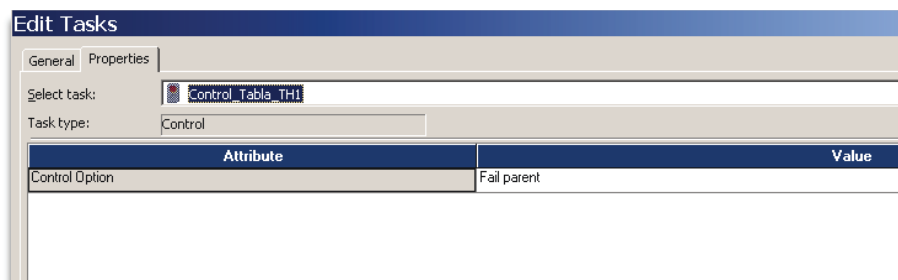


Figura 64: Configuración del Control

En caso de todas las sesiones hayan acabado con éxito, la última tarea es el *Command* que se encarga de crear un fichero de texto llamado "flag_fin_ODS_Ventas.txt" en un directorio especificado en la línea de comandos, con el fin de avisar que la ejecución del workflow ha terminado con éxito.

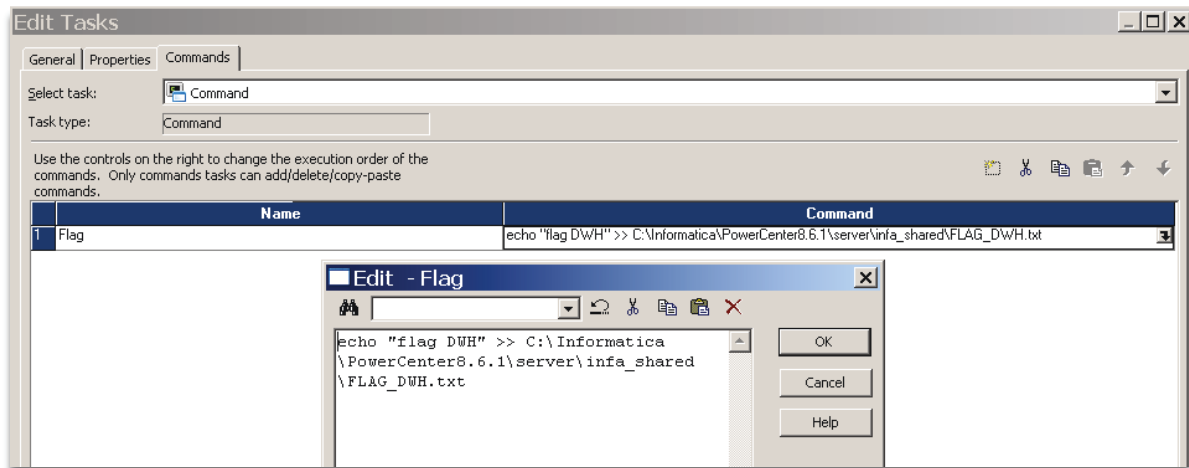


Figura 65: Tarea Command

Finalmente podemos visualizar la ejecución del workflow con todos sus componentes. Podemos observar que todas las sesiones han terminado con éxito (Estado *succeeded*) así como el orden de ejecución de las sesiones a través de las flechas.

De esta manera conseguimos ejecutar 3 sesiones en paralelo que extraen datos de la misma tabla fuente y los guardan en la tabla particionada de manera simultánea en mucho menos tiempo de lo que consumiría hacerlo de la manera convencional (una sesión que realiza toda la operación).

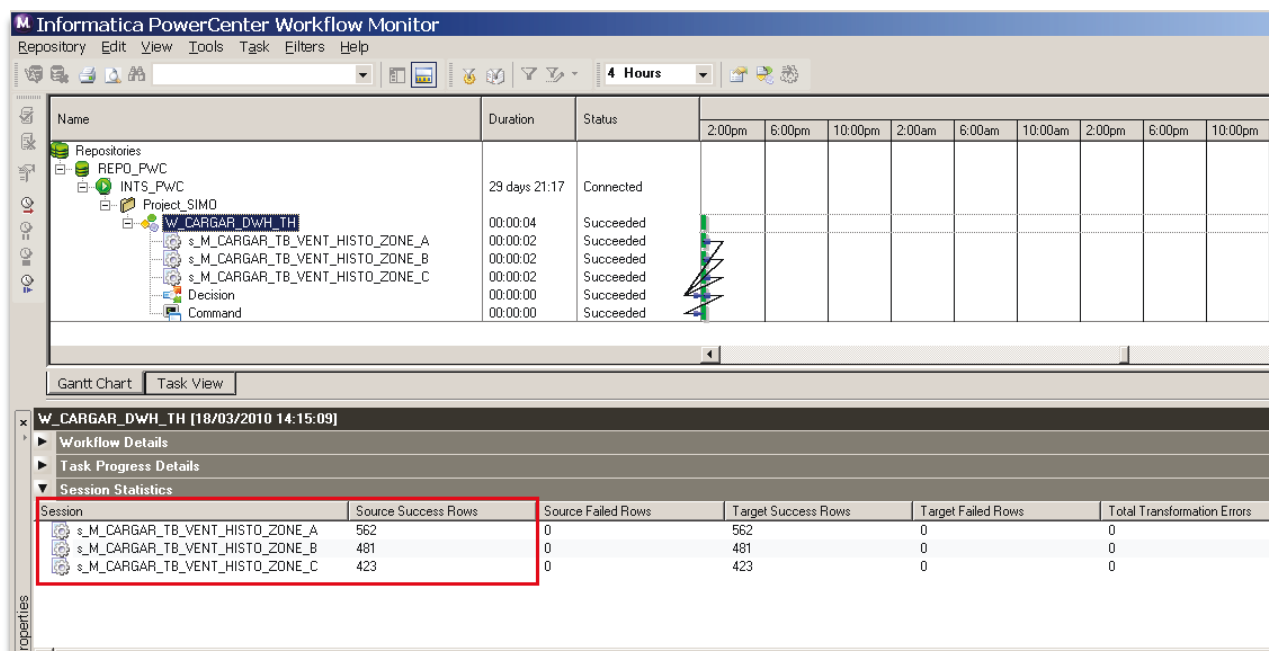


Figura 66: Monitorización del workflow w_cargar_TB_ODS_TH

4.4.3. Fase 3.

Hasta ahora hemos alimentado la tabla que guarda el histórico de datos en la fase de ODS, sin embargo ésta no será accesible desde otra aplicación de Reporting o visualizada directamente por el negocio. En esta fase se diseña un workflow que extrae de la tabla histórica únicamente los datos del mes corriente y los guarda en una tabla final por lo que necesitamos filtrar los datos de manera que no tengamos que realizar cambios en el workflow en cada carga.

Para ello utilizaremos un parámetro que contendrá la fecha que se desea extraer. Este parámetro estará en el fichero de parámetros que se podrá modificar antes de cada ejecución sin necesidad de modificar el workflow.

Este workflow contendrá una única sesión que extraerá los datos de la tabla histórica particionada. Teniendo en cuenta el gran volumen de datos que tendrá esta última, utilizaremos una solución para optimizar el tiempo de extracción.

Los workflows de la fase 2 y 3 se lanzan al mismo tiempo. Para respetar el orden de ejecución de los workflows, hemos añadido mecanismos de control:

workflow 1 -> Flag -> workflow 2

La primera tarea que contiene el workflow es "EventWait". Esta tarea espera la llegada del flag que crea el workflow anterior en un directorio especificado.

Para informar al usuario final del estado de la carga, hemos añadido una tarea "Email" para mandar un correo electrónico en caso de que la carga falle, y así poder intervenir a tiempo y corregir el error.

Contenido del workflow "w_CARGAR_TB_FINAL_TF"

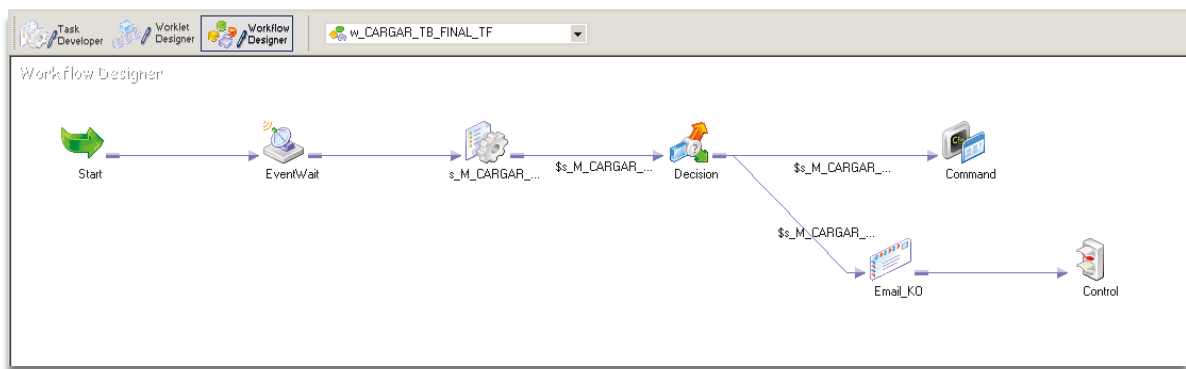


Figura 66: workflow de la fase final

Como podemos ver en la siguiente imagen, la llegada del fichero "FLAG_DWH.txt" termina la ejecución de esta tarea y se ejecuta la siguiente sesión.

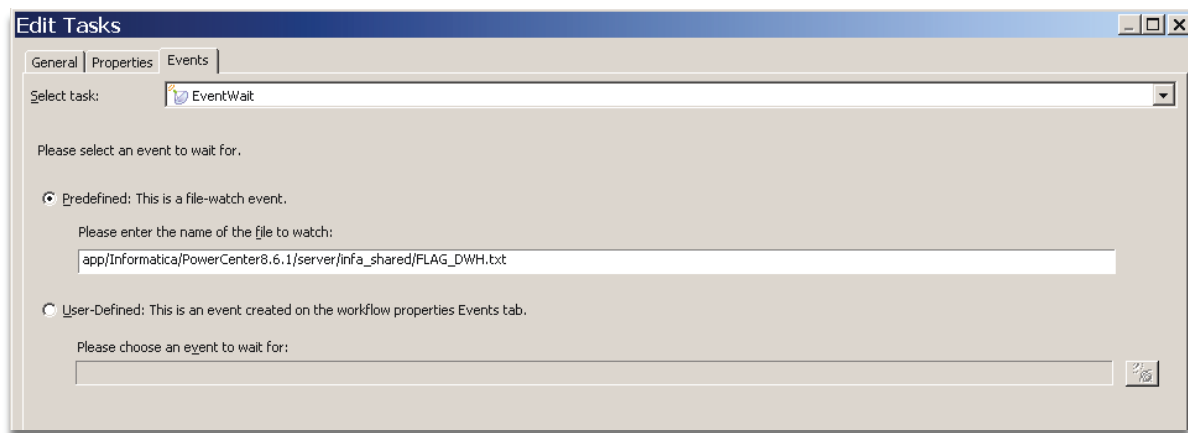


Figura 67: Tarea Wait

Configuración de la sesión que realiza la carga de datos:

En la opción “*Partitions*” añadimos tres particiones en la estructura de la tabla fuente “TB_VENT_ZONE_TH” ya que está particionada. Esto nos permitirá realizar filtros de datos sobre las diferentes particiones de manera separa con el fin de realizar la extracción de datos en paralelo.

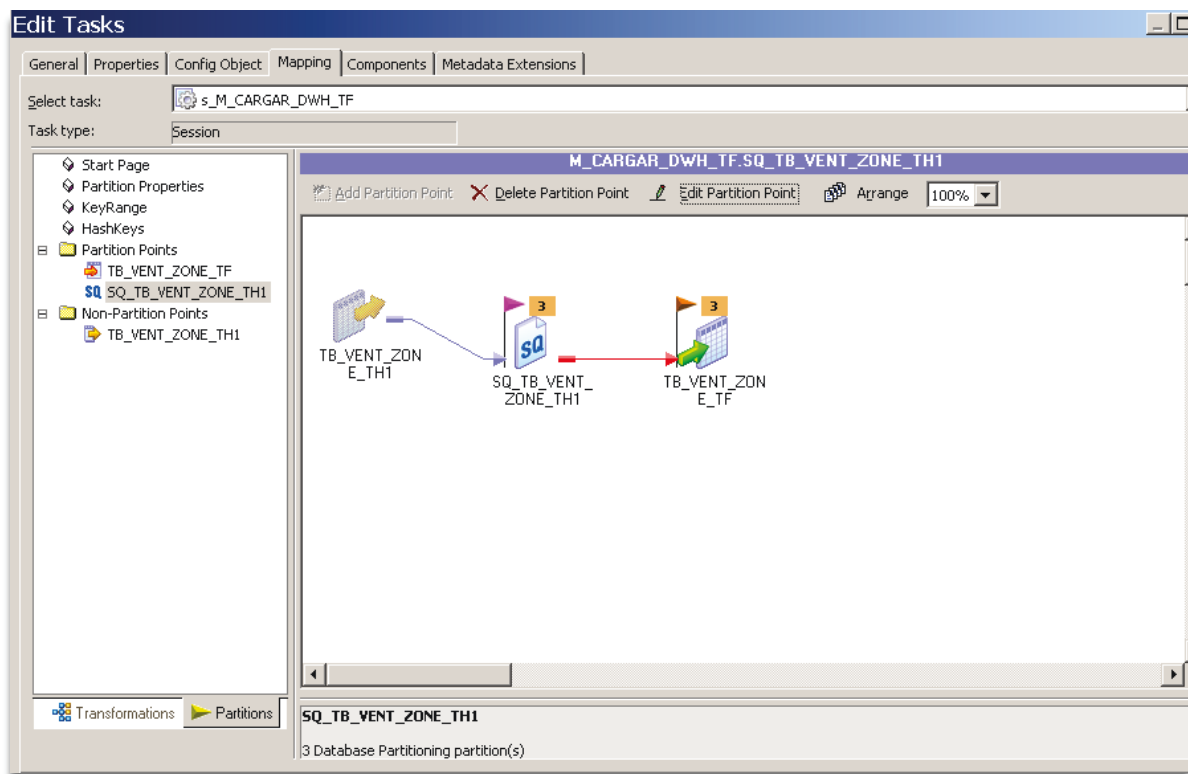


Figura 68.1: Configurar el SQ de la sesión

Imagen de las 3 particiones añadidas a la estructura del SQ.

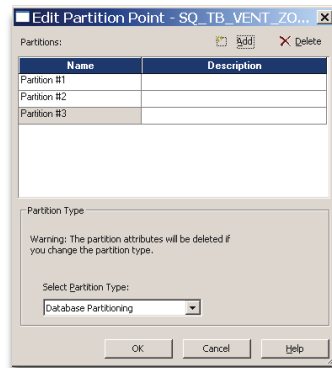


Figura 68.2: Particiones del SQ

Para optimizar el tiempo de carga de datos, ejecutaremos 3 consultas SQL sobre las 3 particiones de la tabla histórica de manera simultánea.

La tabla final tendrá únicamente los datos del mes corriente, por lo que filtraremos los datos en cada una de las particiones utilizando un parámetro “\$\$Fecha” que se encuentra en el fichero de parámetros.

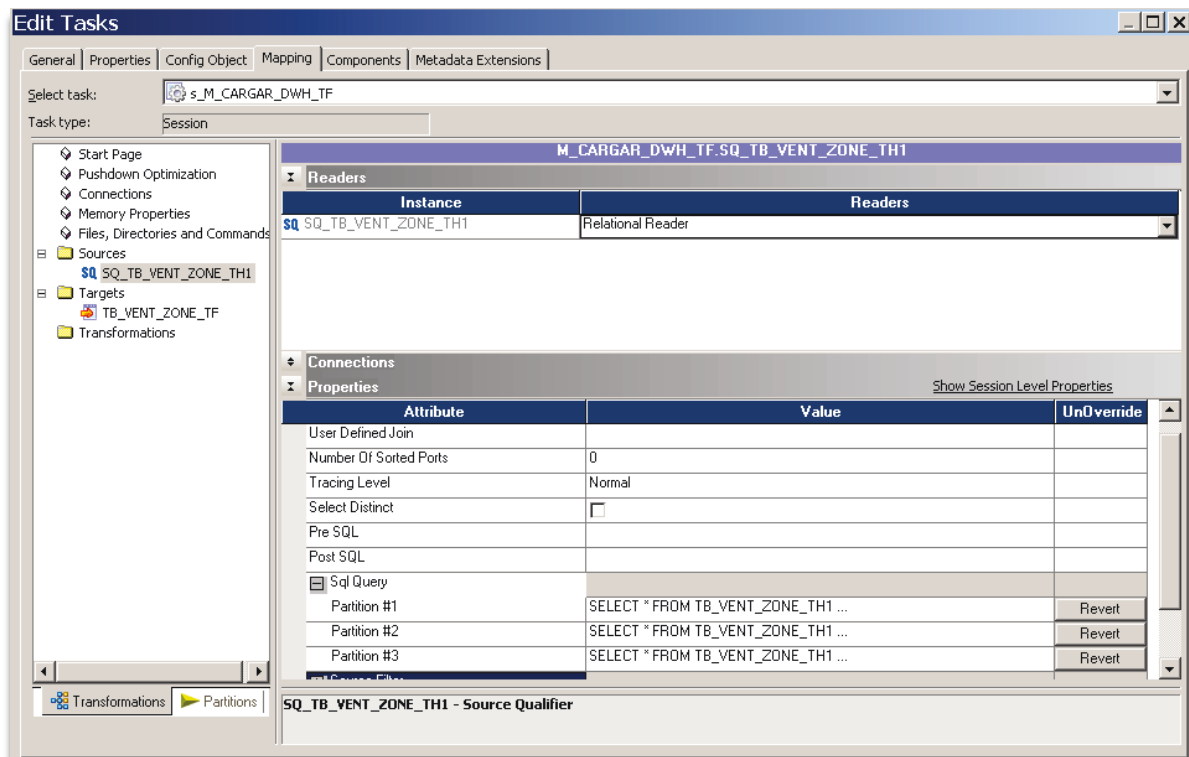


Figura 68.3: Sesión que carga de datos en la tabla final

Cada una de las particiones tendrá su correspondiente consulta SQL:

Partición 1:

```
SELECT * FROM TB_VENT_ZONE_TH1
WHERE TB_VENT_ZONE_TH1.ZONE_ID = 'A' AND TO_CHAR(REPORTING_DATE, 'YYYYMM') =
TO_CHAR(TO_DATE($FECHA, 'YYYYMM'), 'YYYYMM')
```

Filtramos los datos por código de zona y fecha (Reporting_Date). La fecha tiene que ser igual al valor pasado por parámetro.

Esta operación se repite en cada partición cambiando el valor del campo ZONE_ID = 'B' o 'C'.

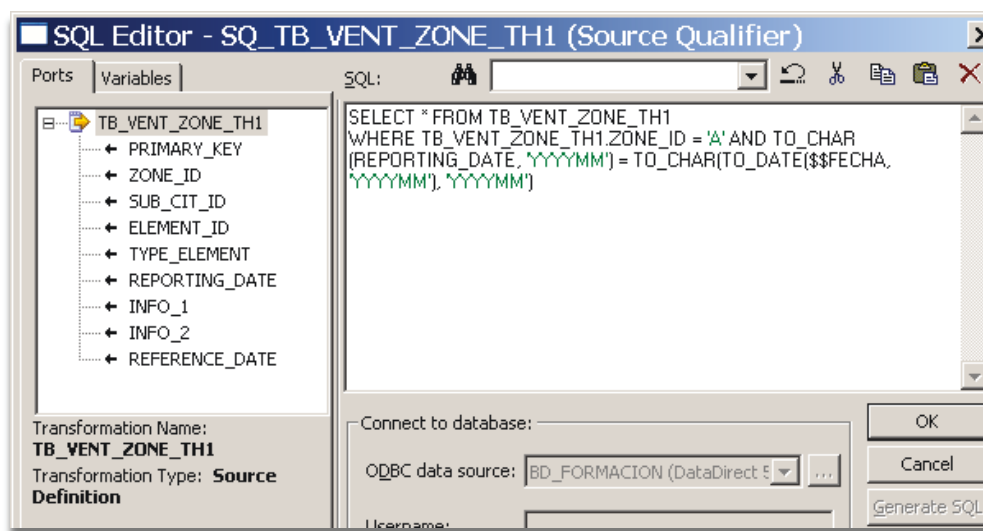


Figura 69: Query de la partición A.

En la configuración del workflow asignamos el fichero de parámetros y la ruta donde se encuentra.

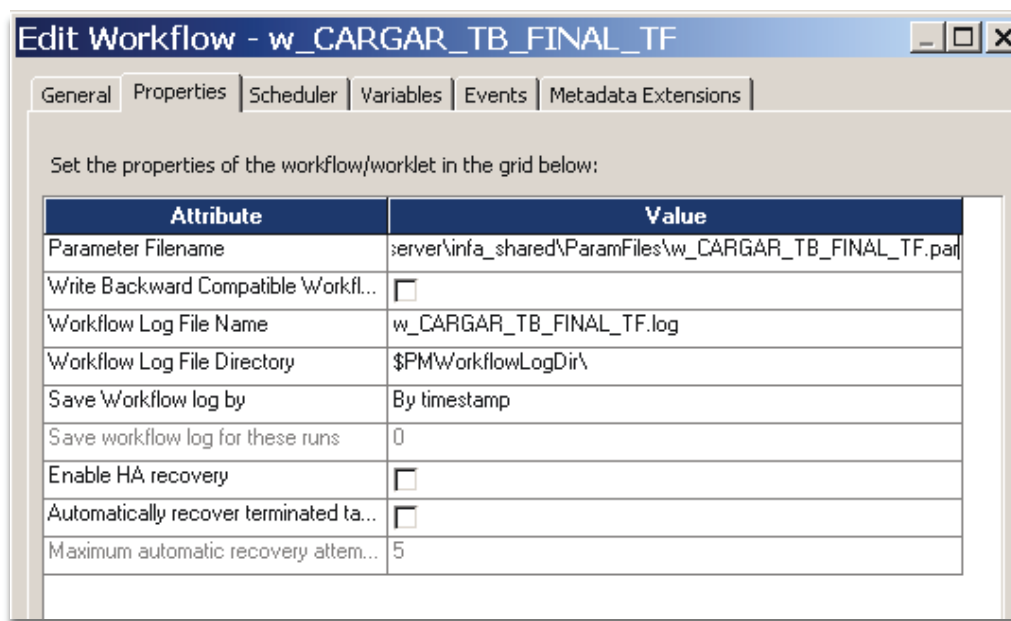


Figura 70: Configuración del fichero de parámetros.

El contenido del fichero de parámetros es el siguiente:

```
[Project_SIMO.WF:w_CARGAR_TB_FINAL_TF.ST:s_M_CARGAR_DWH_TF]
```

```
$$FECHA='201509'
```

A la salida de la Decision, el primer flujo tendrá la condición de la ejecución exitosa de la sesión anterior y así ejecutar la última tarea *Command* que crea el flag informando del final de la carga.

El segundo flujo tendrá como condición `$s_M_CARGAR_DWH_TF.Status != succeeded` indicando el fallo de la sesión anterior. Seguidamente se ejecuta la tarea *Email* que envía un correo electrónico.

Por último se ejecuta la tarea *Control* para abortar el workflow.

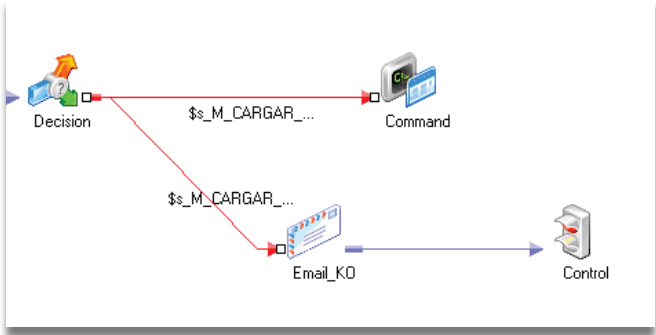


Figura 71: Salida de la Decision.

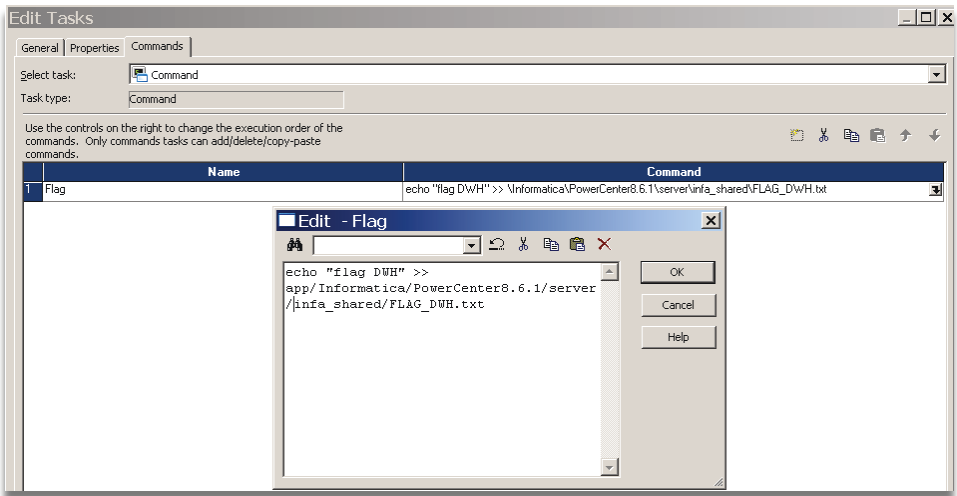


Figura 72: Creación del flag final.

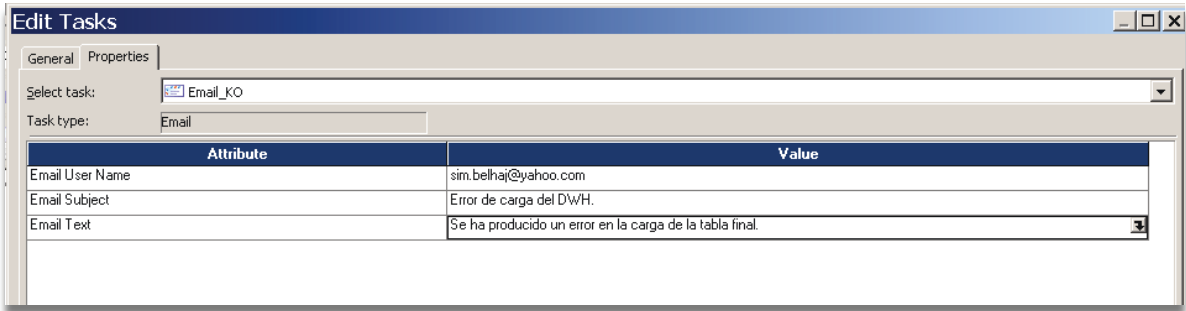


Figura 73: Configuración de la tarea Email.

Finalmente podemos observar en la imagen siguiente la ejecución exitosa del workflow. En las propiedades de ejecución de la sesión, se observa el volumen de datos extraídos en cada partición así como el resto de tareas que contiene el workflow.

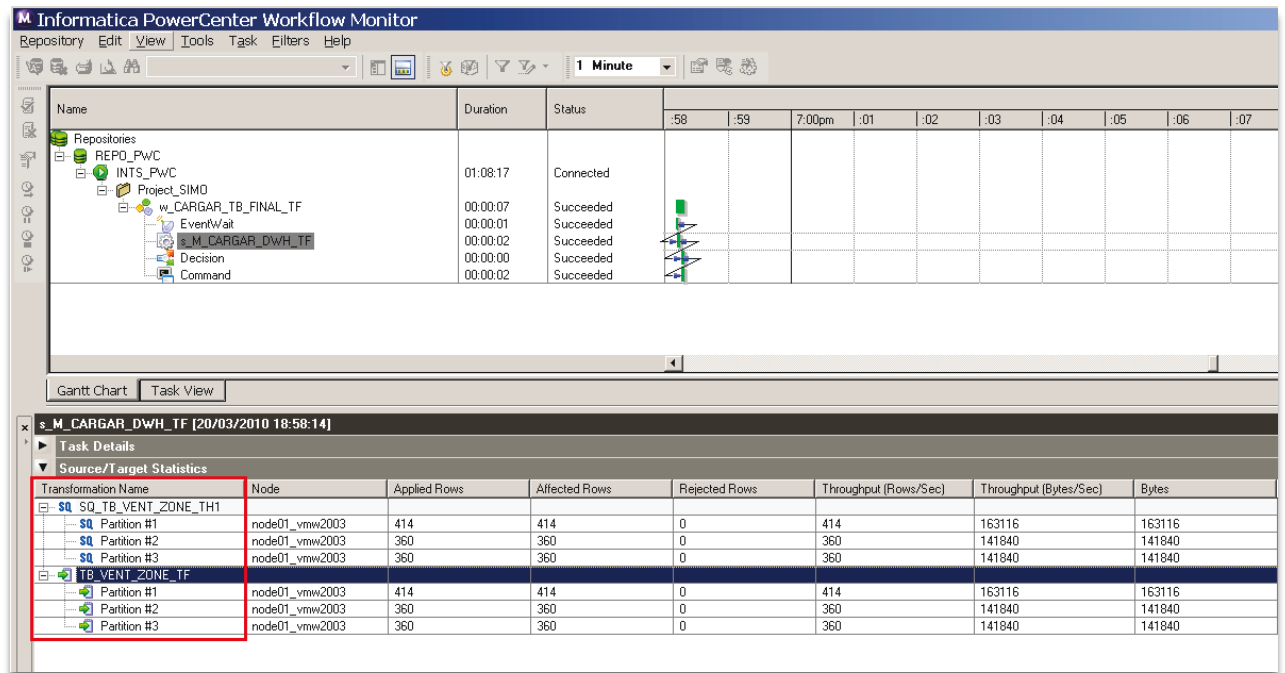


Figura 73: Monitorización del último workflow.

5. Conclusiones y trabajos futuros

5.1. Conclusiones.

Una vez finalizado el desarrollo, podemos concluir que hemos conseguido todos los objetivos marcados en un principio, así como los requisitos funcionales y técnicos detallados anteriormente.

Después de la descripción del funcionamiento de la herramienta y sus diferentes utilidades, hemos intentado implementar un caso práctico basado en requisitos reales que se suelen dar en la mayoría de los proyectos de ETL. Se han propuesto e implementado soluciones a los problemas potenciales que se dan en este tipo de procesos.

Los procesos se han diseñado y desarrollado de manera que optimicen el tiempo de su ejecución aprovechando el particionamiento de la tabla histórica y alimentar el DWH para ser explotado por otras herramientas de Reporting.

Se han incluido mecanismos de control en los procesos de ETL con el fin de controlar el flujo y el orden de ejecución así como remontar posibles errores que pueden interrumpir la carga de datos.

A nivel personal puedo indicar que este proyecto me ha servido para mejorar mis destrezas en el diseño de procesos ETL y sobre todo intentar encontrar soluciones a algunos requisitos de rendimiento. También puedo destacar el aprendizaje obtenido del estudio de mercado que he realizado sobre las diferentes herramientas que hay en el mercado y la demanda que tienen en el mercado laboral.

5.2. Trabajos futuros.

Como cualquier tecnología de hoy en día, los procesos de ETL se enfrentan cada vez a nuevas problemáticas y necesitan evolucionar de forma muy rápida para adaptarse a las exigencias. Un entorno a tener muy en cuenta es el Big Data.

Cuando un proceso ETL tradicional se enfrenta a grandes volúmenes de datos de entrada, una frecuencia de actualización alta en origen o bien son datos no estructurados, estos procesos suelen tener problemas en su adaptación, lo que supone un problema de rendimiento costoso.

Algunas herramientas Big Data utilizan un modelo "Schema on-read" que no necesita una estructura de datos previamente definida y no realiza validaciones en el tiempo de carga de datos. En este caso la estructura de datos se define en el momento de lectura, lo que aporta una gran flexibilidad y rapidez.

Se está viendo que cada vez hay más necesidad de tratar grandes volúmenes de datos y las empresas diseñadoras de herramientas ETL están en una carrera hacia la innovación y adaptación al Big Data. También me gustaría resaltar la gran demanda de ingenieros con conocimientos de Big Data, lo cual abre una importante oportunidad en el mercado laboral.

6. Anexos

6.1. Coste del proyecto

Para la buena gestión de cualquier proyecto hace falta realizar estimación del tiempo de desarrollo así como el coste. En este proyecto, el desarrollo de los procesos ETL se puede llevar a cabo simultáneamente, a diferencia de las dependencias para su ejecución.

Vamos a realizar un presupuesto de los procesos solicitados por el cliente, teniendo en cuenta que el coste y la disponibilidad de otros componentes como licencias, servidores, etc. esta fuera de nuestro entorno y nos limitamos al desarrollo.

Antes de empezar el desarrollo se necesita hacer un estudio funcional y técnico para comprobar los requisitos y la posibilidad de implementar soluciones. Cada tipo de tarea la realiza un perfil determinado y con un coste correspondiente.

En primer lugar veremos los costes por perfiles y por día de trabajo. Un día de trabajo son 8 horas de trabajo que pueden ser realizadas en un día natural o por horas.

Perfil	Tarifa
Jefe de proyecto Junior	420 €
Jefe de proyecto Senior	570 €
Jefe de proyecto Experto	689 €
Analista funcional Junior	370 €
Analista funcional Senior	490 €
Analista funcional Experto	588 €
Analista técnico Junior	320 €
Analista técnico Senior	395 €
Analista técnico Experto	450 €
Desarrollador Junior	305 €
Desarrollador Senior	370 €
Desarrollador Experto	420 €
Arquitecto técnico Junior	430 €
Arquitecto técnico Senior	590 €
Arquitecto técnico Experto	745 €

En la siguiente tabla se detallan las diferentes tareas que se deben llevar a cabo y los perfiles correspondientes. Cada tarea tiene una complejidad con una carga en días (8horas), multiplicada por la tarifa del perfil obteniendo el coste.

Al final se añade la carga correspondiente a la documentación, la asistencia a la hora de la puesta en Producción y el seguimiento del jefe de proyecto.

Tarea	Perfil	Complejidad	Carga	Tarifa	Coste
Revisión de la concepción funcional	Analista funcional Senior	Baja	2,00	490,00	980,00
Concepción técnica	Analista técnico Senior	Media	3,00	395,00	1185,00
Creación de Scripts Oracle	Analista técnico Senior	Media	1,00	395,00	395,00
Creación de Script Shell	Analista técnico Senior	Alta	2,00	395,00	790,00
Creación de Datamap	Desarrollador Senior	Baja	0,50	370,00	185,00
Creación de workflow 1 fase 1	Desarrollador Senior	Baja	1,50	370,00	555,00
Creación de workflow 2 fase 1	Desarrollador Senior	Media	2,50	370,00	925,00
Creación de workflow fase 2	Desarrollador Senior	Alta	3,50	370,00	1295,00
Creación de workflow fase 3	Desarrollador Senior	Alta	3,50	370,00	1295,00
Tests de integración	Analista funcional Senior	Media	3,00	395,00	1185,00
Total Unid.					8.790,00 €

Documentación	15%	1318,50
Asistencia de puesta en Producción	7%	615,30
Pilotaje (Jefe de proyecto)	20%	1758,00
Total Global		12.481,80 €

6.2. Planning

En este apartado planificamos la realización de las tareas del proyecto, teniendo en cuenta las dependencias que hay entre las mismas.

La siguiente tabla podemos ver que hay algunas tareas que se pueden realizar en paralelo.

Tareas	s10					s11					s12					s13				
	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V
Revisión de la concepción funcional																				
Concepción técnica																				
Creación de Scripts Oracle																				
Creación de Script Shell																				
Creación de Datamap																				
Creación de workflow 1 fase 1																				
Creación de workflow 2 fase 1																				
Creación de workflow fase 2																				
Creación de workflow fase 3																				
Tests de integración																				
Entrega de los packages																				

6.3. Scripts Oracle

Añadimos los scripts de creación de tablas Oracle.

- Tabla TB_VENT_ZONE_TH1 histórica:

```
CREATE TABLE "ODS_OWN"."TB_VENT_ZONE_TH1"
  ("PRIMARY_KEY" VARCHAR2(20 CHAR) NOT NULL ENABLE,
   "ZONE_ID" VARCHAR2(1 CHAR) NOT NULL ENABLE,
   "SUB_CIT_ID" NUMBER(3) NOT NULL ENABLE,
   "ELEMENT_ID" VARCHAR2(110 CHAR),
   "TYPE_ELEMENT" VARCHAR2(50 CHAR),
   "REPORTING_DATE" DATE,
   "INFO_1" VARCHAR2(110 CHAR),
   "INFO_2" VARCHAR2(50 CHAR)
   PRIMARY KEY ("PRIMARY_KEY") ENABLE
 ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS NOLOGGING
 STORAGE(
   BUFFER_POOL DEFAULT )
 TABLESPACE "EXAMPLE"
 PARTITION BY LIST ("ZONE_ID")
 (PARTITION "ZONE_ID_A" VALUES ('A')
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
   STORAGE(
     BUFFER_POOL DEFAULT )
   TABLESPACE "EXAMPLE"
  NOCOMPRESS
 ,
  PARTITION "ZONE_ID_2" VALUES ('B')
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
```

```

    STORAGE(
    BUFFER_POOL DEFAULT )
    TABLESPACE "EXAMPLE"
    NOCOMPRESS
,
    PARTITION "ZONE_ID_3" VALUES ('C')
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
    STORAGE(
    BUFFER_POOL DEFAULT )
    TABLESPACE "EXAMPLE"
    NOCOMPRESS
) ;

```

- Tabla TB_VENTAS_AUX del ODS:

```

CREATE TABLE CURSO.TB_VENTAS_AUX
(
    PRIMARY_KEY      VARCHAR2(20 CHAR)          NOT NULL,
    ZONE_ID           VARCHAR2(1 CHAR)            NOT NULL,
    SUB_CIT_ID        NUMBER(3)                  NOT NULL,
    ELEMENT_ID        VARCHAR2(110 CHAR),
    TYPE_ELEMENT      VARCHAR2(50 CHAR),
    REPORTING_DATE    DATE,
    INFO_1            VARCHAR2(110 CHAR),
    INFO_2            VARCHAR2(50 CHAR)
)
TABLESPACE EXAMPLE
PCTUSED      0
PCTFREE      10
INITRANS     1
MAXTRANS     255
STORAGE      (
                INITIAL          64K
                MINEXTENTS       1
                MAXEXTENTS       UNLIMITED
                PCTINCREASE       0
                BUFFER_POOL       DEFAULT
            )
NOLOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

```

- Tabla TB_VENT_ZONE_TF final del DWH:

```
CREATE TABLE DWH.TB_VENT_ZONE_TF
(
  PRIMARY_KEY      VARCHAR2(20 CHAR)          NOT NULL,
  ZONE_ID          VARCHAR2(1 CHAR)            NOT NULL,
  SUB_CIT_ID       NUMBER(3)                  NOT NULL,
  ELEMENT_ID       VARCHAR2(110 CHAR),
  TYPE_ELEMENT     VARCHAR2(50 CHAR),
  REPORTING_DATE   DATE
)
TABLESPACE EXAMPLE
PCTUSED          0
PCTFREE          10
INITTRANS        1
MAXTRANS         255
NOLOGGING;
```

6.4. Scripts Shell

Los siguientes códigos corresponden a los scripts desarrollados en Unix y que tiene como misión lanzar la ejecución de los workflows en el orden establecido utilizando variables de entorno previamente definidas.

- Script "Lanzar_carga.sh":

Este script contiene el orden de ejecución de los workflows. Los workflows de la primera fase se ejecutan secuencialmente ya que utilizan la misma tabla destino. El script controla la ejecución de cada uno devolviendo cada el valor de salida 0 o 1.

Los workflows de la segunda y tercera fase se lanzan simultáneamente, pero no hay riesgo de desorden en la ejecución ya que se han integrado mecanismos de control por *flag* en los propios workflows (tareas *Command* y *Wait*).

Este script llama a los siguientes scripts que realizan la ejecución.

```
#####
# declaración de variables
#####
```



```
fecha=`date "+%Y%m%d%H%M"`

ruta_shell="/home/.../bin"
ruta_log="/home/.../tmp/Log_Carga_Historica_${fecha}.log"

#####
## Lanzar la ejecución del primer workflow de la fase 1. ##
#####
${ruta_shell}/exec_workflow.sh Project_SIMO w_Extract_Cob_Files

echo "Project_SIMO w_Extract_Cob_Files" >> $ruta_log

retorno_shell=$?
if [ ${retorno_shell} -ne 0 ]
then
    echo "Probleme workflow Project_SIMO.w_Extract_Cob_Files" >>
    $ruta_log
    exit 1
fi

#####
## Lanzar la ejecución del segundo workflow de la fase 1.##
#####
${ruta_shell}/exec_workflow.sh Project_SIMO w_Extract_Flat_Files

echo "Project_SIMO w_Extract_Flat_Files" >> $ruta_log

retorno_shell=$?
if [ ${retorno_shell} -ne 0 ]
then
    echo "Probleme workflow Project_SIMO.w_Extract_Flat_Files"
    >> $ruta_log
    exit 1
fi

#####
## Lanzar la ejecución del workflow de la fase 2.##
#####
${ruta_shell}/exec_workflow_nowait.sh Project_SIMO
w_CARGAR_TB_ODS_TH

echo "Project_SIMO w_CARGAR_TB_ODS_TH" >> $ruta_log

retorno_shell=$?
if [ ${retorno_shell} -ne 0 ]
then
    echo "Probleme workflow Project_SIMO.w_CARGAR_TB_ODS_TH" >>
    $ruta_log
    exit 1
fi
```

```
#####  
## Lanzar la ejecución del workflow de la fase 3.##  
#####  
{ruta_shell}/exec_workflow_nowait.sh Project_SIMO  
w_CARGAR_TB_FINAL_TF  
  
echo "Project_SIMO w_CARGAR_TB_FINAL_TF" >> $ruta_log  
  
retorno_shell=$?  
if [ ${retorno_shell} -ne 0 ]  
then  
    echo "Probleme workflow Project_SIMO.w_CARGAR_TB_FINAL_TF"  
>> $ruta_log  
    exit 1  
fi
```

- Script "exec_workflow":

Este script realiza la ejecución de un workflow sin permitir otra ejecución hasta que no se haya acabado el anterior.

```
# Comprobación de parámetros  
[ $# -eq 2 ] || { what $0; exit 1; }  
  
CARPETA=$1  
WORKFLOW=$2  
  
# Ejecución del workflow  
cd $RUTAINFORMATICA/.../bin  
pmcmd startworkflow -sv $I_SERVICE -d $DOMINIO -u $USUARIO -p  
$PSSWD -f $CARPETA -wait $WORKFLOW  
salida=$?  
  
exit $salida
```

- Script "exec_workflow_nowait":

Este script realiza la ejecución de un workflow permitiendo otra ejecución posterior sin que haya acabado el anterior.

```
# Comprobación de parámetros  
[ $# -eq 2 ] || { what $0; exit 1; }  
  
CARPETA=$1  
WORKFLOW=$2  
  
# Ejecución del workflow  
cd $RUTAINFORMATICA/.../bin
```

```
pmcmd startworkflow -sv $I_SERVICE -d $DOMINIO -u $USUARIO -p  
$PSSWD -f $CARPETA -wait $WORKFLOW  
salida=$?  
  
exit $salida
```

7. Bibliografía

- Documentación Informatica Powercenter :
 - <https://www.informatica.com/es/products/data-integration/real-time-integration/powercenter-real-time.html>
 - <https://community.informatica.com/onlinehelp/analyst/961/es/index.htm#page/release-guide/GUID-53B0D9AA-1D11-4F23-9D02-D59152984AB6.1.005.html>
- Documentación de base de datos Oracle 11g:
 - https://docs.oracle.com/cd/E11882_01/nav/portal_5.htm
- Documentación de PowerExchange:
 - <http://international.informatica.com/la/products/data-integration/powerexchange/>